Roundup: Best beginners' distro

ON YOUR FREE DVD

Elementary OS Luna, Hotpicks, Pinguy OS 13.04 64-bit + more!

*Get the ISO of the DVD from www.linuxformat.com/downloads

LINUX FORMAT

#1 for Free Software

Fund your dreams

How crowdfunding is transforming the economy of free software

» Get the backing you need
» Synergise the interwebs
» Become your own boss!

6 pages of programming excellence

CODING ACADEMY

Fortran Hack like it's 1957 p88
Python Print hello world p84

Inside Qt

» How one tiny graphical toolkit took the world of mobile development by storm p46

IS YOUR FREE DVD MISSING? Speak to your newsagent now!

Renaissance man

"Most people today write more than Charles Dickens did in his lifetime"

Robert 'r0ml' Lefkowitz on, well, everything p42

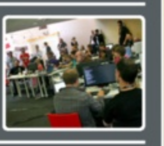
Also inside...

Ubuntu 13.10

Has Ubuntu lost it/still got it/got it back/always been ace? p18

PyCON UK 2013

Parselmouths dwell in the New Model stronghold of Coventry p50



Qt 5

What next for Nokia's lost orphan child? p54

Desktops
Customise a GUI
» Build a bespoke desktop environment

Websites
Drupal
» Unleash world-beating Web 3.0 sites the easy way

Load balancing
HAProxy
» Beat the Slashdot effect with high availability

Future
TuxRadar.com

LINUX FORMAT
#1 for Free Software

What we do

Corporate Patron Free Software Foundation

» We support the open source community by providing a resource of information, and a forum for debate.

» We help all readers get more from Linux with our tutorials section – we've something for everyone!

» We license all the source code we print in our tutorials section under the GNU GPLv3.

» We give you the most accurate, unbiased and up-to-date information on all things Linux.

Who we are

In this issue, we're celebrating the crowdfunding revolution. We asked our contributors what project they'd like to see on Indiegogo or Kickstarter...

Chris Thornett
Mine's already there – it's *Hyper Light Drifter* by Heart Machine!

Andrew Gregory
For consistency, I'm going to say a Linux port of *Elite: Dangerous.*

Efrain Hernandez-Mendoza
How about a local RepRap 3D printing community in the Bristol area?

Ben Everard
Having ridden a wooden scooter across Africa, I'd simply ask for an off-road map.

Mayank Sharma
How about something similar to the Ubuntu Edge, only with a target of $12m?

Jonathan Roberts
My dream is to create the perfect pasty. What's the minimum target?

Mike Saunders
After writing my own x86 operating system, I'd like to fund a new version written in Forth.

David Cartwright
Could someone start a campaign to build a cheap yet awesome Linux laptop for the UK?

Nick Veitch
I'd like to buy the rights to *Blade Runner 2*, so that Ridley Scott can't destroy it.

Juliet Kemp
I'd create a project to fund open source driver creation to help the Nouveau folks out.

Shashank Sharma
Space exploration should be crowdfunded!

Neil Bothwick
Graham says, "One thing I will miss is making up these answers on deadline day."

New beginnings

Working on this magazine has always been a dream come true. You get to write about the subject you love and talk to many inspirational people within the Linux and free software community. Prior to this job, I'd been messing about with open source software for a few years, eventually developing an application called *kalbum* that made its way into a few distros, and I wanted to find a way of turning that passion into a career. I spotted the staff writer position advertised on the *Linux Format* website, and applied.

Back in 2004 the magazine was run by Nick Veitch, the erstwhile editor of the legendary Amiga Format magazine. Having been both an Amiga addict and an avid reader in the 90s, I was a little starstruck at meeting and working with Nick, especially as we were interviewing Stephen Tweedie (of ext3 filesystem fame) within days of me starting. But it didn't take long to get settled into the lack of routine and the pressure of deadlines. We've all come a long way since then.

But it's time to move on, and I'm sorry to report that this will be my last issue here at Future Publishing. Andrew Gregory and Ben Everard have also decided to leave, so it's something of a new beginning at Linux Format Towers. I've had some wonderful times here, from interviewing Linus Torvalds and breaking the Raspberry Pi story, to challenging expectations with our 'Learn to Hack' and 'Beat the CIA' features, all of which I'm very proud of, especially in the light of recent surveillance revelations.

Taking over editorship is Neil Mohr. Neil has written for, edited, and worked on many magazines over a long career in technology journalism, and I'm sure he'll do a first-rate job here at *Linux Format*. I'm looking forward to seeing what he and the team do with such an esteemed title!

Graham

Graham Morrison Editor
» Follow me on Twitter **@degville**

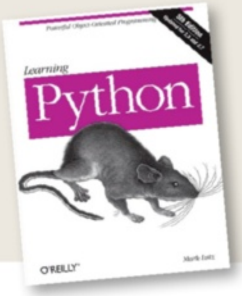
Contents

We are all in the gutter, but some of us are looking at the stars…

Reviews

› **New search lenses make it into Ubuntu 13.10 – but there's no Mir yet.**

› **Darkness falls across the land. The midnight hour is close at hand.**



Add your voice to the

Sound of the crowd

You can go your own way with the power of crowdfunding p34

Roundup: Beginners' distros p26

Talking heads

> We're at that cusp where we're starting to say 'oh, we should teach everybody how to code'

Robert 'R0ml' Lefkowitz AD 800 and all that p42

LINUX FORMAT DVD DISTROS | SOFTWARE | GUIDES

Elementary OS Luna

The best-looking distro you'll try this year

Pinguy OS 13.04

Want to get started with Linux? Here's the distribution for complete beginners

64 bit

LXF178 Plus Books, code, podcasts, Hotpicks and more!

Open Index.html to start exploring

On your free DVD

Elementary OS Luna

» Recommended by podcast listeners!

Pinguy OS 13.04

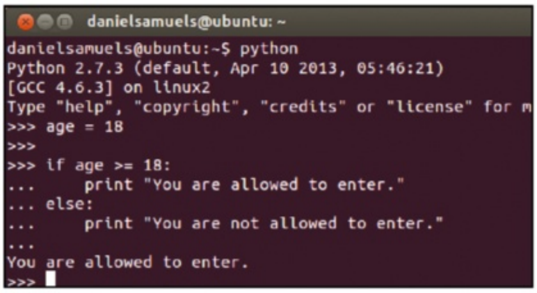
» Get started with Linux

PLUS: HotPicks and tutorial code p96

Treat yourself or a loved one to an LXF subscription! p32

Don't miss...

Coding Academy

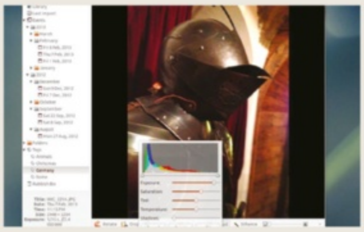


```
danielsamuels@ubuntu:~$ python
Python 2.7.3 (default, Apr 10 2013, 05:46:21)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for m
>>> age = 18
>>>
>>> if age >= 18:
...     print "You are allowed to enter."
... else:
...     print "You are not allowed to enter."
...
You are allowed to enter.
>>> 
```

Regulars at a glance

› **Our subscriptions team is waiting for your call.**

Tutorials



› ***Shotwell*: even better than *Kalbum*.**

LINUX FORMAT Newsdesk

THIS ISSUE: A million Pis » AbiWord » Cinnamon » Qt Developer Days » Web hosting

EVENT

Show report: Qt Developer Days

Graham Morrison joins 600 other attendees in Berlin catching up with the *Qt* folks.

› **The Kino International. Home to Cold War propaganda and the 2013 *Qt* Developer's Days.**

This is the tenth year of *Qt* Developer Days, and the second under the auspice of KDAB and Digia, both of whom bought Nokia's stake in the platform last year. Why does *Qt* still matter? Well, it's the toolkit used to drive the KDE desktop, as well as hundreds of other applications. And the great advantage that *Qt* has over its rivals is that, as well as being open source, it's a commercially supported product. This gives *Qt* a professional sheen, especially when it comes to documentation and stability (*see What On Earth, p54 for more details*).

It's been a great 12 months for the platform, especially from a Linux perspective, as many new developers are considering *Qt* as an alternative to *GTK*. In recent weeks and months, projects such as *Wireshark*, *OpenShot* and LXDE have announced their intentions to switch, mostly because of the excellent cross-platform advantage. Add to this the renewed breadth of the API by encompassing so many smartphones, something *Qt* wasn't properly able to develop under Nokia, and the entire framework is looking stronger than it has done for years. We can't imagine what life would have be like had *Qt* still been part of Nokia and consequently bought by Microsoft, but we're glad it's now in the hands of people who understand and appreciate its unique advantages.

Which is why we found ourselves once more in Berlin just as the leaves were turning golden along the avenues. The keynotes themselves were held in the Kino International cinema, directly opposite Cafe Moskau, which was the location for the remainder of the sessions. The cinema is a marvelous throwback to harder times, complete with parquet tiling, brown and green carpet on the walls and the smell of a million Cold War cigarettes. This is a much better venue than the Munich Hilton favoured by Nokia, and the expansion to the cinema itself a sure indication that *Qt* is growing.

Qt comrades

The propaganda comparisons start early, as Matthias Kalle Dalheimer, President and CEO of KDAB, takes to the stage. He reminds us how the cinema was used to show premieres to the GDR government, sat in the front row, teaching the populace about its "eternal and unbroken friendship with the Soviet Union." He goes on to joke this may be similar to the relationship he has with the *Qt* community, " Thank you, comrades," he chides before getting more on-topic with attendance statistics. The focus is obviously on a decade of *Qt*, with growth in the two digits and the proclamation that "we're still growing." The breakdown of attendees is interesting, and is unlike many of the other conferences we attend. 31% of attendees are doing so for the first time, while 33% are in their second year. That's a lot of new *Qt* users, which must surely be down to

› ***Qt 5.3* includes *Wayland* support for the desktop, a cross-platform packaging solution and App store integration, as well as advanced QML profile debugging.**

the renewed emphasis on *Qt* becoming equally adept at working across smartphone platforms as it is on the desktop. Android is talked about a lot, as too is a port to iOS and the planned release of a Windows RT port. Blackberry is named-dropped as the platform with the best *Qt* support, but that may be more to do with the long running strategic alliance between the two companies than an objective opinion on what works best and where.

Android support

Tommi Laitinen, SVP at Digia, was next to take to the stage. It was his job to reassure desktop users they weren't being abandoned in the race for mobile supremacy, while also trying to appeal to the rapidly growing group of mobile developers who obviously wanted the best cross platform experience they could get their hands on. Maybe this is why revenues grew by 30% in the first half of this year and why Tommi's own presentation was running off an Android tablet. You certainly couldn't tell the difference between its output and that of a laptop, although he kept transitions to a minimum and there was a distinct lack of cats.

> **"Lars announced Qt is going to switch to yet another web engine."**

Tommi also made the point that his Android presentation app was the same as the app running on a desktop. *Qt 5.2* is in alpha, with a beta due soon and a final release scheduled before the end of the year. Version 5.2 is the target for a real cross platform solution, and the release where Android becomes a first-class citizen and will be fully supported for the first time. This is badly needed, as one of the sessions we attended covered only the basics of creating a build environment for Android, and that was a tough hour to follow.

Tommi also announced perhaps the most contentious addition to *Qt*'s products for quite some time, the *Qt* Mobile Edition. This is a subscription service, costing $149 per month per user, giving them the ability to build both iOS and Android versions of their app. Of course, there's nothing wrong with paying for support and the package also includes cloud storage for your projects, both for development and for your clients. But it's the way apps are deployed on iOS and Android that's worrying, as a commercial mobile edition seems to hint at some elements of the project becoming closed source. Your app will probably need to be statically linked to *Qt*'s iOS library, for example, which may not be possible with the open source edition. We weren't able to get the definitive answer on this at the event.

This led on to Lars Knoll, now the CTO and Chief Architect at Digia. He explained there are now fewer developers at Digia than there were at Nokia, but that there's more freedom for them to work on things that are important to them. He even admits to being able to code himself again, after a long period spending 90% of his time doing admin at Nokia. He also admits to being sceptical about the release of 5, after so much of the architecture was changed, but he's happy with the results, and especially, it's compatibility with earlier versions. The tech on the embedded side is being driven by the wide adoption of touch screens, and this is something *Qt* needs to improve on. 'Boot to *Qt*' was also announced. This is an embedded version of *Qt* that features a complete stack to take your hardware from the bare metal to a *Qt* hosting operating system. But it's closed source, with no plans of an open source release. This may be forgivable when anyone is free to do the same thing using LGPL *Qt*, and it will probably only be of real interest to well heeled real-time vendors such as QNX anyway.

Just after announcing that *Qt 5.1* shipped with the best version of *WebKit* (a framework Lars himself kickstarted with KHTML), Lars announced *Qt* is going to switch to yet another web engine. But he's got good reasons. *WebKit* development under Apple isn't very open. Google, by comparison, is much happier to work with third-parties and platforms, and it's forked *WebKit* into *Blink* for similar frustrations. Lars remarked that when they saw Google's roadmap for *Blink*, it was the first time they'd seen a web engine roadmap. For these reasons, *Qt* is going to switch to the same engine for the 5.3 release, due in the first half of next year. iOS users will be stuck with *WebKit*. We're looking forward to see some similar thawing in *Qt*'s wider adoption (and in KDE 5!) over the next 12 months.

Newsbytes

» Nvidia has removed a Linux-specific feature from one of its drivers, to drag us down to the same level as its Windows customers. The BaseMosaic feature used to enable Linux users to use four screens, but this feature has been disabled, Nvidia says, "for feature parity between Windows and Linux".

» Apparently Dick Cheney asked his doctors to switch off his heart defibrillator's Wi-Fi capability in case terrorists used it to give him a heart attack. You can read more about the implications of this in our interview with the Gnome Foundation's executive director Karen Sandler, the full text of which is online at **www.tuxradar.com**.

» Another year, another OggCamp. The unconference celebrated its fifth iteration in English city of Liverpool, home to Echo And The Bunnymen, and has become the largest free software and free culture event in the UK. And, like FOSDEM in Brussels, it's still free to attend.

› Liverpool: European City of (Free) Culture. Also home to The Teardrop Explodes and The La's.

» Google has expanded its bug-hunting bounty, offering cash prizes to anyone who makes security updates to OpenSSL, OpenSSH, BIND, zlib and other crucial bits of internet plumbing. The bounties range from $500 to $3,133.70.

» The latest Ubuntu release, Saucy Salamander, has shipped without the promised Mir display server, leading to an unseemly row in which Ubuntu founder Mark Shuttleworth compared opponents of Mir to the US Republican Party's Tea Party wing. In other news, the Gnome 3.10 release includes experimental support for the Wayland display server.

HARDWARE

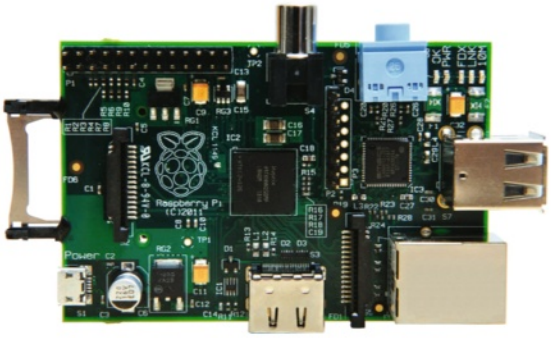
Raspberry Pi passes million milestone

Over a million Pis have poured forth from Pencoed.

The Raspberry Pi Foundation has released the fantastic tidbit that it has now made over 1 million of its tiny PCs at its British manufacturing facility.

The brains behind the Foundation have always intended the device to be made in the UK, but initially had to have it made in China. Once the Pi had launched, it became clear that demand was high enough to justify switching production to Sony's factory in Pencoed, South Wales, and the millionth unit rolled out of the factory in early October 2013.

Eben Upton, Raspberry Pi founder, told the BBC "I remember being told this was an unsaleable product. But we've already surpassed the sales of the BBC Micro – my childhood computer. There was a latent need for something like this."



› Brains beer, castles, Raspberry Pis, the National Health Service – Wales is the gift that keeps on giving.

If you're interested in learning to program with the Raspberry Pi – or teaching someone else to do it – look out for Ben Everard's forthcoming book *Learning Python With Raspberry Pi* from O'Reilly.

ADOPTION

French police migrate 37,000 PCs to Linux

Slow and steady wins the race for our continental cousins.

The French Gendarmerie has released an update from its ongoing project to migrate its desktop computers from Windows XP to Linux, and it's all *très bon*.

In contrast to other, rushed schemes that have gone awry, the French approach has been one of incremental adoption, with users first making the transition to free software applications, including *Firefox* and *LibreOffice*, before switching to a custom version of Ubuntu. The project is already bigger than the city of Munich's migration project, which involved a mere 14,000 desktops.

Major Stéphane Dumond of the French Ministry of the Interior told the Evento Linux conference in Lisbon that, "Using an open-source desktop lowers the total cost of ownership (TCO) by 40% in savings on proprietary software licences and by reducing costs on IT management. Using Ubuntu Linux massively reduces the number of local technical interventions." Addressing the question of total cost of ownership, Major Dumond added: "The direct benefits of saving on licences are the tip of the iceberg. An industrialised open source desktop is a powerful lever for IT governance."

› When the migration is complete, all 72,000 of the Gendarmerie's PCs will run desktop Linux.

WORKS WITH MySQL
php
TYPO3
Windows
1&1
WordPress
Joomla!

Comment

In search of identity

Tony Chapman

So, I'm going to attempt to tackle the perhaps slightly contentious issue of DevOps in the recruitment space, but only briefly as I'm not sure I could handle the potential backlash!

For a recruiter it's potentially a nightmare, simply because there appears no standard definition of what DevOps do. Every company we speak to has a different interpretation of what they require and it turns out to be very subjective for a search campaign. In literal terms it is obviously development and operations together, but skills-wise we tend to look for areas like automation, CI/CD, cloud, configuration management; someone who can bridge the gap between the two skill sets. The main question is: does DevOps Engineer as an actual job title exist?

No boundaries

I'm not going to go into too much depth about the ins and outs about what 'real' DevOps is or could be; my point is that the many people I speak to each day get very uptight about the term DevOps. 'We've been doing DevOps before they gave it a name' they say, 'It's just a new name for Sysadmins to earn more money' or 'company x claim to be doing DevOps however they are actually not', etc. I just feel they should relax slightly – much like the person who gets so uptight about whether tomato ketchup should be room temperature or in the fridge – seriously, does it really matter that much?

No two companies have ever had the same way of working, even before DevOps. Good engineers and developers have always placed their own interpretation on methodologies and trends. DevOps is still a relatively informal way of working with no clear set boundaries and is open to interpretation – so why not just run with it or work to define it?

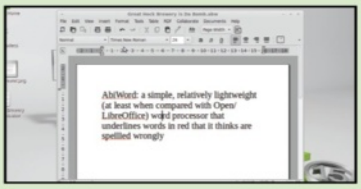
› Tony is a leading Linux and open source recruiter at specialist agency LinuxRecruit.
www.linuxrecruit.co.uk

Hitting the mirrors

What's behind the free software sofa?

OFFICE SOFTWARE

We love *LibreOffice*, but it's still a bit too much for some older, or low-powered machines (try it out on a Raspberry Pi and you'll be stuck there for ages waiting for it to load). Luckily, for older machines, or for anyone who just wants a simple, clean interface,there's *AbiWord 3.0*. This is the first major release of AbiWord since 2.8 all the way back in 2009, and the biggest change is that it's now built using *GTK 3* rather than the old 2.x version of the graphical toolkit. There are the usual bugfixes and tweaks, but the concept – a word processor that does one thing, well – is unchanged.


File Edit View Insert Format Tools Table ADF Collaborate Documents Help

AbiWord: a simple, relatively lightweight (at least when compared with Open/ LibreOffice) word processor that underlines words in red that it thinks are spellled wrongly

› If you don't like visual distractions while you're writing, try *AbiWord*.

LINUX DISTRO

The latest stable version of Red hat Enterprise Linux, 5.10, was released at the end of September, bringing with it essential updates for server admins with the budget to pay for a Red Hat's support contracts.

For everyone else (or just for home dabblers who want to learn a *bona fide* enterprise grade Linux distribution) there's CentOS 5.10. CentOS aims to provide a 100% binary compatible version of RHEL, and as such it's the best way to teach yourself how to use the distribution of choice for thousands of your potential employers.

Formatting / file system...
CentOS-5
Community ENTerprise Operating System

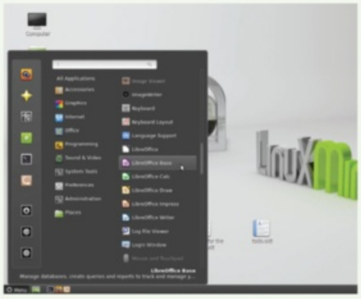
› CentOS 5.10 still uses the same *MySQL* versions as Red Hat does.

LINUX DISTRO

Mandriva's slow decline brought a tear to our eye. But, thanks to the fact that it's free sofware, anyone can come along and rescue the source code. And that's exactly what has happened! OpenMandriva Lx 2013 beta is now upon us, and with it we have our first glimpse of this community continuation of the old Mandriva distro. The distro uses KDE 4.11, and has "a focus on a clean and unified desktop". We can't wait to compare it with the other Mandriva offshoots, Rosa and Mageia.

DESKTOP

Cinnamon, the fabulous desktop environment developed by the Linux Mint team, has released a new version. This was previously a front-end to Gnome, putting back much of the functionality that the Gnome devs had stripped away in the name of simplicity. Cinnamon 2.0 is a standalone environment, however, though it does still use *GTK* and Clutter, for example.



› Cinnamon 2.0 brings sound effects and improved tiling for large screens.

United Linux!

The intrepid **Les Pounder** brings you the latest community and LUG news.

Find and join a LUG

- **Nottingham LUG** 7pm, every first and third week. Visit the website for more details. **http://nottingham.lug.org.uk**
- **Greater London LUG** Details can be found on the website. **www.gllug.org.uk**
- **Tyneside LUG** From 12pm, every first Saturday of the month at the Discovery Museum, Blandford Square, Newcastle. **www.tyneside.lug.org.uk**
- **RaspberryJam Manchester** Monthly at Madlab. Visit the website for more details. **http://madlab.org.uk/content/manchester-raspberry-jam-3**
- **Manchester Hackspace** Open night every Wednesday. Visit the website for more details. **http://hacman.org.uk/**
- **Lincoln LUG** Third Wedsnesday of the month at 7pm, Lincoln Bowl, Washingborough Road. **http://lincoln.lug.org.uk**
- **Hull LUG** 8pm at Hartleys Bar, Newland Ave, first Tuesday every month. Visit the website for details. **http://hulllug.org/** for more details.
- **Surrey & Hampshire Hackspace** Third Tuesday of the month. More details on the website. **http://sh-hackspace.org.uk**
- **Preston Hackspace** Every two weeks on a Tuesday at The Continental. Details on the website. **http://prestonhackspace.org.uk/**

Drop-in and solder away

LUG Blackpool gets a makerspace

The Blackpool LUG has been part of many great things in the past few years: the group regularly travels around the UK to help at events, such as Oggcamp, Raspberry Jams and the Barcamp Blackpool. But just recently, the LUG has achieved something fantastic: it's successfully created the first makerspace in the Blackpool area.

The team has worked to build a place where anyone can drop in and use it free of charge and there are resources and equipment to tackle most types of project. The LUG is currently playing with Arduino, Raspberry Pi and Fignition, and are investigating its own Raspberry Pi powered PBX phone system. Mike Hewitt, the LUG master for Blackpool LUG is currently busy working on a soldering and welding area, and has extended the opening hours to accommodate the increasing demand.

Mike has been the LUG master since the mid-2000s, and has spent a considerable amount of time and resources making the LUG and now the makerspace a success. Mike runs a PC recycling, community focused business, and the LUG is very lucky to have access to such a great space for free.

The makerspace isn't quite a complete hackspace yet, but it is a perfect fit for the needs of the group, with electronics and hardware hacking being the main activities.

If you're in the area, why don't you pop in for a coffee and a chat soon? Or if you can't make it, why not drop in to its Google Hangout every Saturday morning from 10am. You can find the Blackpool LUG and makerspace details at **http://blackpool.lug.org.uk**. LXF

Photo credit: Les Pounder

LUG Blackpool gets stuck into many events around the country and now has a makerspace.

Community events news

Photo credit: Nicolas Vigier

FOSDEM 2014
The massive open source conference is back for 2014, between February 1-2. The fabulous city of Brussels in Belgium will play host to 2,000 hackers, system administrators and linux users, all coming together for a weekend of scheduled talks and workshops.

One unusual aspect of FOSDEM is that you will be able to take your LPI (that's the Linux Professional Institute) certification exams during the event. If you're looking to gain Linux certification, and fancy a few excellent Belgian beers, this is the place to be. You can find out more about FOSDEM, by visiting the website. **https://fosdem.org/2014/**

Electromagnetic Field 2014
This is a massive event that takes place – yes, you've guessed where – in a big field. Think of it as a makerfaire, hackspace, LUG, conference and social gathering, all rolled into one amazing place. There's even going to be an underground bar, selling various microbrewery ales. The price of admission is well worth it, for this unique and exciting event. You can find out more details from the official website. **www.emfcamp.org**

M2M Hackfest
If you are interested in the internet of things, ubiquitious computing, ambient intelligence, haptic echnology then the M2M (Machine to Machine) Hackfest is for you. It's a great chance for you and a team of people to build new innovative technologies, with lots of coding and hacking. The event takes place November 26-27, at the Olympia, London and the full programme of events can be found on the website. **http://the-hackfest.com**

Mailserver

Write to us at *Linux Format*, Future Publishing, 30 Monmouth Street, Bath BA1 2BW or **lxf.letters@futurenet.co.uk**

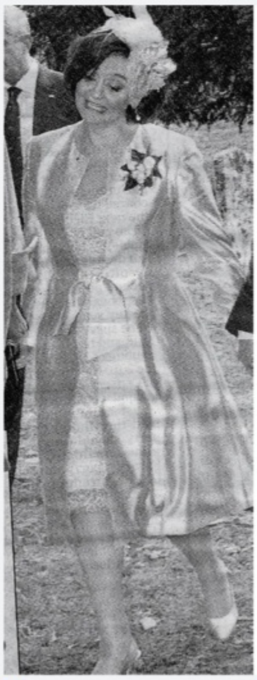
Letter of the month

A touch of Cherie Blair

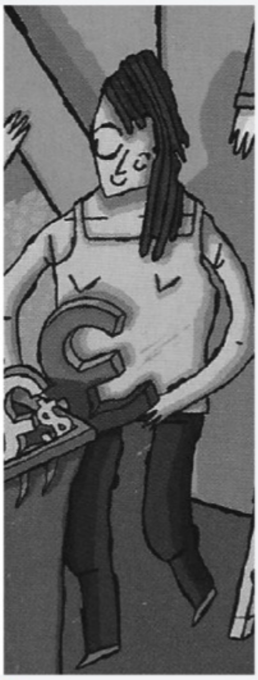
Hallo, and thanks for keeping me interested in linux & LXF. I've often wondered where you got the inspiration for your cartoon mannikin from. Now I know!

Alan Newman

Andrew says: We applaud the likes of Cherie Booth Blair QC for her crusading efforts in promoting religion as a mitigating factor in assault cases and for injecting much-needed capital into the Bristol housing market, helping to keep plebs in rented accommodation where they belong. I for one welcome our cartoon overlords.



Cartoon character.



Cherie Booth Blair QC.

Plus ca change

I enjoyed the Doctor's piece about 'Mary' in LXF177. In the 1970s I was helping an old lady in the village with some forms and got in touch with the council on her behalf. I gave her name, address and some other details they wanted, and when asked for her telephone number replied: "She doesn't have one." Silence, then in a response in tones of disbelief, "What do you mean she doesn't have a telephone?"

I can't remember, but I don't think I explained to the young council official that my friend had been born at the end of the last century (19th) and had lived in her cottage all her life. At some point a cold water tap had been installed in the kitchen but there was no electricity, no hot water, the toilet was an outside, earth privy and she did her washing once a week in an old fashioned copper in a shed at the bottom of the garden. Plus ca change, plus c'est la meme chose!

dmk

Andrew says: Even when I were a lad in the 1990s there were kids at school who didn't have a telephone in the house. If there's a lesson here it's that technology should be an enabler rather than a limiter. If you choose to communicate only in one way, you're limiting yourself and your customers/taxpayers, which is sub-prime. If you have a Facebook page instead of a website, or if you only listen to customer complains that come from Twitter rather than a phone line, you're not leaving luddites and technophobes behind – you're cutting yourself off.

LINUX FORMAT DVD
DISTROS | SOFTWARE | GUIDES
MINT 15 Mega Mix
64-bit only
Cinammon
Mate
Unity
KDE
Bodhi
A minimal new wave distro
Siduction
Boot to the ultra fast Razor-qt desktop
LXF174
Plus: Books, code and more!
Open Index.html to start exploring

Sometimes, Mike does good things. This was one of those times.

Viruseserii

Not so long ago, you featured the AntiVirusLive CD. I burnt the ISO image to CDR 'just in case'. Well, this weekend became the 'just in case' weekend. A friend's daughter was having problems with her laptop. During this adventure I used the AntiVirusLive CD to get the laptop into a suitable state so I could sort out other problems and install Microsoft Security Essentials. Thanks for the really useful DVD!

Ian

Andrew says: I looked for ages at the front covers of the DVD wallets and didn't find any mention of AntiVirusLive CD. Then I realised that the ever-excellent Mike Saunders had included this brilliant little find in Hotpicks. You clever sausage Mike! Even though there are no viruses for Linux out in the wild, antivirus software still has a place on Linux – if only so we can help our friends who are afflicted with Windows viruses.

Tor haven

I've just been listening to your podcast (Series 5 E14) and in particular your discussion on the compromising of Tor. One of your team trotted out the usual "if you've done nothing wrong why worry". Well, the problem is, who's deciding what's right or wrong. For example, if you live in, say, China, entering Tiananmen Square as a search word in your browser could get you a visit from The Man. If you live in, say, Uganda and live an 'alternative lifestyle' and try to network with like-minded people you could face a death sentence. We are fortunate in the UK that we live

› British mathematician, computer scientist and war hero, sitting next to the statue of Alan Turing in Manchester.

in a more liberal society and don't have to worry about getting the Midnight Knock, but for many around the world the integrity of a system like Tor enables them to exercise what little liberty the have.

"They who can give up essential liberty to obtain a little temporary safety deserve neither liberty nor safety". Benjamin Franklin
Enjoying the mag as ever.
Reg

Andrew says: Hey Reg, that was Graham, our outgoing editor. He was just being silly, which is why we all laughed at him as soon as we said it; we know that online anonymity is important, which is why we were talking about *Tor* in the first place.

But you raise an essential point. The 'I've got nothing to hide' fallacy is used all the time by people who should know better. It's all very well for us to laugh at Graham, but, as you rightly say, lots of people do have things to hide, and many of them have perfectly legitimate reasons for doing so.

› Perhaps the flipped rotation in the southern hemisphere accounts for Australia's constant difficulties against spin?

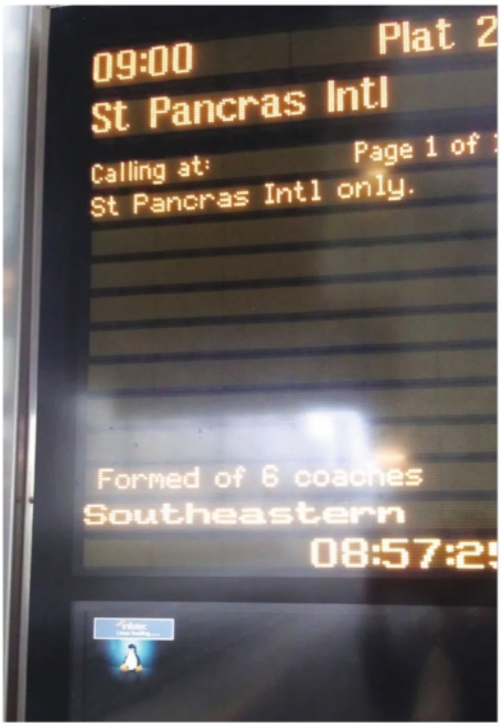
G'day, Bruce

You have probably been asked this before, but you know those 'wait for it' icons known as throbbers? Well, do they rotate counterclockwise in the Southern Hemisphere?
Chuck (Long time reader).

Andrew says: Good question. I smell a lucrative research grant – Bondi here I come!

Station to station

I am an avid subscriber to your rag and listener to your podcasts. Thought you might like to see this 'awesome' snapshot my girlfriend (who despite my best efforts is yet to migrate to the Linux desktop) just sent me from Stratford International station.
Jason



› It warms our cockles to see Tux providing essential infrastructure support tasks.

A mint release

I have tried Mint over a number of years but have always returned to my favourite, Xubuntu – until Mint 15. The guys at Mint have really got things right this time as far as I am concerned.

A friend had bought a new laptop with Windows 8 and hated it. I co-installed Mint 15 from the coverdisc on LXF174. I installed *boot-repair* to get it to boot and my friend is now happy and Microsoft have lost yet another user.

I was so impressed that I co-installed it on my Windows 7 laptop and then Microsoft came to my aid. *Disk2vhd* from the emporium at Redmond is a piece of software to back up and create a VHD file from a laptop or server. Using this software, I backed up my laptop onto a 1TB USB drive, switched the laptop to Mint 15 and fired up Oracle VirtualBox, then created a new machine, selected Use An Existing Disk – the VHD file on my USB drive that I had created earlier. Boot it up, install *VirtualBox* additions and it all works just like a native Windows installation. I had to reboot the virtual machine a zillion times but that is normal for Windows. The laptop has 4GB of memory and 4 CPU cores. I allocated half of both to my virtual machine. Wonderful!
Mike Ryder

Graham says: For anyone who has yet to try Mint 15, we can't agree with Mike strongly enough: the guys at Mint have really got it right. The installation process is easy, and flexible enough to enable even a novice user to create a separate, encrypted/home partition.

As a desktop, Cinnamon makes sense, and thanks to Mint's Debian underpinnings it's easy to find and install as much free software as you'll ever need. You can install codecs and restricted drivers with one click, it has a pretty login screen, and we even love the file manager.

Firefox vs Chrome

Thanks for your interesting comparison, but I do not agree with your scores at all as *Chrome* is missing just one thing for world domination – just one. Many people are clamouring for it and the *Chromium* devs and Google are not listening.

The 'missing' feature? It's a Bookmarks sidebar! I have loads of bookmarks nested within folders, and far too many to use in a bookmarks toolbar. I would switch to *Chrome* in a second if it had this.

Here's a tip for you... try typing in about:flags into the Chrome address bar for a nice little surprise.
Guy

Graham says: I wouldn't be too surprised if someone has read this and is now working to create a Bookmarks sidebar just for you.

After all, that is the beauty of open source.

Safe and sound?

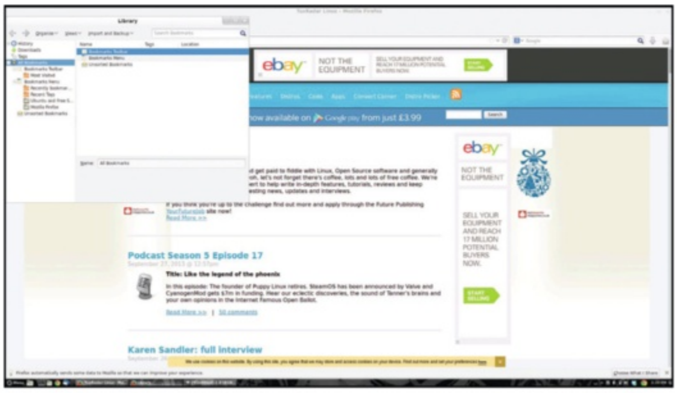
I've been using Ubuntu for a while and I've found it much better then Windows. One of the reasons that brought me to Linux is my perception that with this system I will be able to protect my privacy and get a higher level of security compared with Windows. I feel that this is something quite important today when hardly a day passes without revelations about some form of cyber-espionage, pursued either by American NSA or other governments or simply by plain old criminals.

I'd like my system to be secure and safe, so among other things, I'm trying to follow all the advice that software developers give concerning installation of their packages. One thing that I don't fully understand concerning security is the verification of packages. When for example I want to download *Tor* they explain to me that I should verify the integrity and signature of the package, they mention GPG signatures, other software that I wanted to download says something about MD5 checksums or mentions SHA1 hashes.

What exactly all this means? Should I verify those things each time I'm planning to install a program? If someone does not verify his packages is he making the same mistake as someone who leaves default username and password to his wireless router or chooses 'password1234' for his Facebook account? Can we allow ourselves not to verify packages? Could you explain to me and other readers just what serious consequences this kind of thing can have for a user of Linux?

Pawel Miech

Graham says: Excellent question! Verified packages come from a trusted source, such as Ubuntu's official repositories and not from third-party sites.


Library
Organize
Views
Import and Backup
Name
Tags
Location
ebay
NOT THE EQUIPMENT
SELL YOUR EQUIPMENT AND REACH 17 MILLION POTENTIAL BUYERS NOW
now available on Google play from just £3.99
Podcast Season 5 Episode 17
Title: Like the legend of the phoenix
Read More >> | 10 comments
Karen Sandler: full interview
START SELLING

› *Firefox* should have won our versus review against *Chromium* in LXF176 for some readers, because of its support for bookmarks.

DREAMLAND
MY SPARKLING BOOK OF
VEGETABLES
Fedora 19: the LXF verdict
LINUX FORMAT
#1 for Free Software
Raspberry Pi camera module
THE NEXT GENERATION OF SYS ADMIN
FREE DISC
EMPIRE
For Mac, iPad and iPhone users
iCreate
FREE CD
UNC

› £10 on the newsstand? That's madness! Sensible Dubai-dwellers save money by subscribing.

How much!

Since moving to Dubai a few months ago I was very relieved to find I could still buy *Linux Format* in some shops and even though it's just over 10 pounds an issue, it's still worth it because as you can see from the photo the local reading material isn't up to much. Although, I've always thought *Linux Format* could do with more holographic glitter. What does Effy reckon?

Anna McGee, Dubai

Effy says: My Sparkling Book Of Vegetables looks fantastic, but I wouldn't trust it for advice on crowdfunding your own world-beating project, administering a dynamic, content-rich website using Drupal or bashing together your own desktop environment. I loves shiny stuff. My precious…

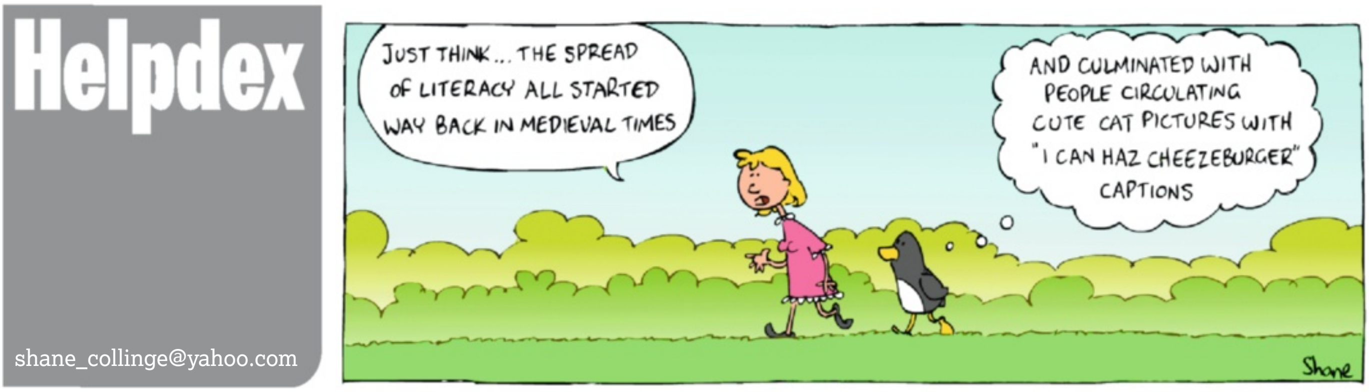
Windows on Linux

It's great to see Linux finally getting some recognition as a gaming platform. An article, or maybe a short series, I would love to see would actually be on the subject of Linux gaming. I would like to see an outline of what options there are eg *Wine*, *PlayOnLinux*, Steam etc and some tutorials on how to use the software. *PlayOnLinux* for example has a scripting facility to get Windows games to work, but how does it work?

Also, the use of Windows software generally on a Linux platform could maybe covered. For example, recently I have managed to get *GotoMyPC* and Sky TV (I wanted Sky Sports) to function in Ubuntu. I had to do it by trial and error because there is so little information out there that I could either find or actually understand. The key to *GotoMyPC*, which I have to use for work, is to use a User Agent and *Wine*. Sky, which uses Silverlight, was solved by use of *Firefox* with a User Agent and *Wine*. An article could cover things like User Agents, *Wine*, *PlayOnLinux* etc, and it would have helped me massively if I had known about them prior to setting out on making Windows software work in Linux.

Richard Austin, Wakefield

Chris says: Some great ideas there for tutorials we can do, Richard! We're got a big feature on Linux gaming in **LXF179**, but I'm mostly covering where we are with native games. Maybe I should do a follow-up on the alternatives? LXF


Helpdex
JUST THINK... THE SPREAD OF LITERACY ALL STARTED WAY BACK IN MEDIEVAL TIMES
AND CULMINATED WITH PEOPLE CIRCULATING CUTE CAT PICTURES WITH "I CAN HAZ CHEEZEBURGER" CAPTIONS
shane_collinge@yahoo.com
Shane

Where's my LXF?

Is Linux Format getting to you late? For some, it seems so, but for the most part the Royal Mail is still doing its job efficiently. Long live this treasured public resource!

Early bird

I saw the letter in the most recent November issue about a subscriber complaining about receiving the magazine late. I have a subscription in the USA. I have always received the issue very early in the month. It's 5 October and I received my November issue (LXF176) yesterday. I think 5th day of the month is the latest I've received an issue since subscribing. Thanks for being so great, I'm loving this month's issue and I'm barely into it.

Keith Shannon, Oakland, California

Learning Linux early

In response to Steve Cox's Mailserver letter (*Where's my LXF?* **LXF176**), I can only say that my subscription copies arrive well before the month printed on the cover. I'm reading **LXF176** for November, which I received in the last week of September. I know that, when I used to purchase the issues from a local bookstore, they would frequently arrive somewhat later. I think if Steve Cox subscribes, he won't have any problems.

While I'm at it, I'll thank you for the magazine. I've been reading it for about a year and a half now, a time that has seen me move from Linux Mint to Slackware and now to Debian Testing. I know I've seen people online talking about wanting to "learn Linux," but while the major Linux distros have some things in common, they also have substantial differences, making it hard to 'learn Linux' overall. I'd like to see your magazine focus a little more on the innards of Linux – perhaps with a Gentoo section, a Slackware section, and a Debian section – than on all the random open-source applications that are available.

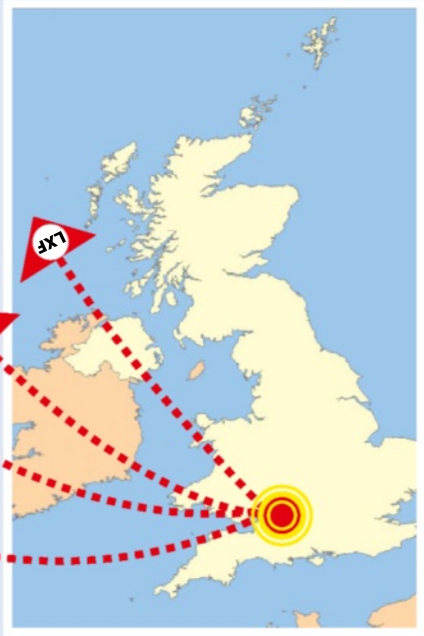
Roy Birk, Maryland, USA

Loyal Les, we salute you

I have been a faithful subscriber to *Linux Format* for over 10 years and only once had a problem receiving an issue late. That was due to Hurricane Sandy, which delayed the mail from the receiving port. Most issues arrive on time.

Les B Labbauf, Harrisburg, PA

Andrew says: It's good to know that, for the most part, the system works and our readers across the pond are getting the magazine in good time. I should point out that we start the year about a month ahead, and, because of the bonus extra special value 13th issue (at Christmas) we end the year about two months ahead, so if the November issue is still managing to reaching the USA in September, something is working properly (*See Fig 1 for details*).



› Fig 1 Here we see the magazine distribution routes of our Avro Lancasters.

Bangkok delayed

Further to Steve Cox's letter in LXF176 which I picked up from reception at my office in Bangkok this morning (23 September) this means it could have arrived on 21 or 22 September.

Something has happened in the past few months where I am not receiving them as quickly as I used to, which has caused embarrassment when I've gone to the my favourite magazines' website to request replacement issues (I've waited in the past for the next issue to be available) only to find the magazine turn up a few days later.

I've been a subscriber since LXF6 and over the years only had a few issues go missing or arrive really late, but now it seems to be a regular occurrence. The first delay occurred after I renewed my subscription for the next three years. I thought I'd missed the deadline to renew. It would be nice to get the magazines in a timely manner, as 5 to 6 weeks after the on-sale date is a bit too long. I know I can download the PDF and read it on my tablet before hand but it's just not the same as having it to hold and flip open in my hand.

Anyway enough of that, thanks for producing a quality magazine I particularly enjoy reading Dr Brown's Administeria, I'm looking forward to getting my teeth into LXF176's Sed tutorial: 18 years of on-and-off Unix/Linux administration and I still can't get my head around it.

Andrew Hubbard, Bangkok

Andrew says: Six weeks to get to Bangkok is pretty lax to me. But that's a long way from the Shire. What shocked me was that another reader, Simon Lynge gets his **LXF** 19 days after it goes on sale, and he's only in Denmark. If I stand on tiptoes I can almost see Denmark from here, so it's baffling that it takes so long.

Keep the queries coming when your copy of *Linux Format* is late – we put a lot of effort into making it, so we want you to enjoy it in a timely fashion.

YAY! I BOOKED THE LAST OF THE 314 AVAILABLE CONFERENCE REGISTRATIONS
GOOD FOR YOU
YEP. NOW I JUST NEED TO PAY THE $314 REGISTRATION FEE
IT'S THE PY-CON, RIGHT?
NO, THE PI-CON. IT'S ON AT THE SAME TIME
Shane

Write to us

Do you have a burning Linux-related issue you want to discuss? Do you simply want to hail Ben's awesomeness or bemoan the distinct lack of PulseAudio criticism in the magazine these days? Write to us at *Linux Format*, Future Publishing, 30 Monmouth Street, Bath BA1 2BW, or **lxf.letters@futurenet.co.uk**.

THE SUNDAY TIMES 100
BEST COMPANIES TO WORK FOR
2005, 2006, 2007, 2008, 2009, 2011, 2012, 2013
FORTUNE 100 BEST
COMPANIES TO WORK FOR
2008, 2009, 2011, 2012, 2013
EMPLOYEES' CHOICE
glassdoor
BEST COMPANIES TO WORK FOR

LINUX FORMAT

Reviews

All the latest software and hardware reviewed and rated by our experts

BEN EVERARD left *Linux Format* last month and is currently cycling across Israel.

Onwards

Along with Graham Morrison and Andrew Gregory, I've left Future Publishing and *Linux Format*, but I certainly won't be leaving Linux.

Not only have I almost finished work on *Learning Python with Raspberry Pi*, a book I'm working on with Alex Bradbury of the Raspberry Pi Foundation, I've finally got around to writing about my cycle adventures across Africa.

It was in Africa where I saw one effect of open source you don't often see in Europe, and that's open source as the only option. For schools, community centres and even businesses, Linux is the only way to go if you don't want to break the law. But it's also the only option if you don't want to pay for support or access to reference materials and tutorials. This is one area where Linux and open source really shines, and we don't often give this facet enough exposure.

We're spending a lot of time in the UK trying to reorganise education, and inspire a new generation to use computers in more creative ways. But perhaps we're not doing enough in other places that are desperate to learn, and there's little infrastructure that I know of to help publicise this.

The Raspberry Pi Foundation has already had a profound effect in many areas other than Europe and the US, and while the emphasis on learning may be different, it still carries the same message – that technology can be cool and liberating, and time spent learning how to interact with open source, Linux and the Raspberry Pi, won't be time wasted.

Our pick of this month's releases:

Ouya: it was at its most popular before people owned one.

Ubuntu 13.10

***Mir* didn't make it into this release, but there's plenty of small refinements that did.**

SYABH

***Sir, You Are Being Hunted* is such a long name that most references are abbreviated.**

Newbie distros p26

Zorin OS

Pinguy OS

Elementary OS

Solus OS

Pear Linux

We've only just noticed that four out of these five distributions have 'OS' in their names! This must be a prerequisite for many distros suitable for beginners.

Ubuntu 13.10

While a new release is always a cause for celebration, with more of the same old, same old in Ubuntu 13.10, **Shashank Sharma** is a little disappointed.

In brief...

» The distro that people love to hate, yet it still remains the most popular Linux distro. See also: Mint, Mageia, OpenSUSE.

While each Ubuntu release is analysed more closely than most scientific discoveries, the latest iteration from Canonical will forever be known, not for what it brought to the table, but what it left out.

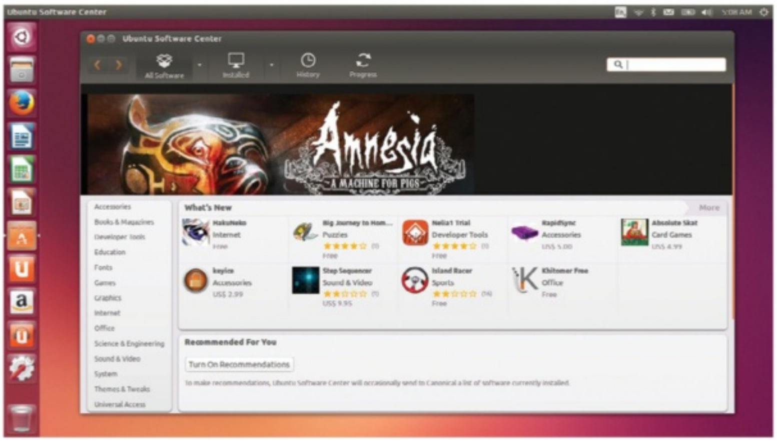
There was a slew of features announced for Ubuntu 13.10, which made it one of the most anticipated releases in a long time. But the Ubuntu team caused an uproar when it subsequently decided that the much awaited *Mir* system – Canonical's built from scratch replacement for the ageing X.org display system, – will not be released as part of the 13.10 release.

Ubuntu 13.10 looks like a casualty of the fixed release cycle as it offers nothing of note and little to distinguish it from its predecessor.

One of the most popular computing acronyms from the 1990s: WYSIWYG feels the most applicable when speaking of Ubuntu 13.10. While the latest release ships with Linux kernel 3.11, a newer Unity, the obligatory bug fixes, and improved performance, most other changes are things that you can see and experience.

> "Dash has transformed into a query-answering powerhouse."

To begin with, the ability to set up and sign into an Ubuntu One cloud storage has been bolted into the installation process.



› The Software Center remains one of Ubuntu's greatest gifts to mankind.

But the biggest talking point for this release, apart from the *Mir* business, is the inclusion of a bunch of new Scopes in Dash. You can still use Dash to search for files, applications and multimedia, but the new Scopes enable you to fetch search results from over 50 online sources, including Amazon, eBay, Etsy, Wikipedia, The Weather Channel, SoundCloud. DeviantArt, Foursquare, and several Google projects like Google Drive, Google News, and Google Books.

Enough scope?

With all these Scopes at its disposal, Dash has transformed into a query-answering powerhouse. You only have to type in the name of your city to get a weather update, or the name of an album or product to get Amazon product listings or Wikipedia entry.

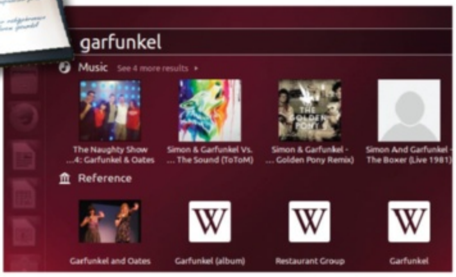
You can also limit the search results by selecting the Scopes you wish to use for any given query. For instance, you can restrict the results to Google News and Wikipedia.

If manually selecting and deselecting Scopes seems like too much effort, you can use prefixes with your queries to specify the source you wish to use. So the prefix **news:** or **wiki:** will only display the latest news or Wikipedia entry for the query term. Unfortunately, prefixes aren't available for all the Scopes and while some are intuitive and self-explanatory, like books for Google Books, wiki for Wikipedia, news for Google News, videos, photos, music, help, and weather etc, such prefixes or modifiers aren't available for all services. Don't bother trying **finance:** or **business:** in the hopes of getting results from Yahoo Finance.

Dash thrives when you use single word search queries but it fails when the query is longer. So, searching for 'Madonna' will overwhelm you with results, but a search for 'David Cameron' will leave you disappointed, regardless of your political leanings.

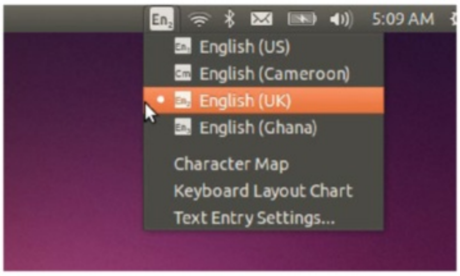
Dash's behaviour when using several words in your queries isn't consistent however, as it sometimes works flawlessly. As with searching on the internet, you can lock your query terms in quotes for more accurate results.

Features at a glance



Smart Scopes

Unity's Dash has had an overhaul under the hood and has tons of new scopes that search over 50 online sources.



Minor Tweaks

This release mostly includes minor tweaks to the distro, such as the new language icon in the status panel.

Unfortunately, with all the Scopes in use, it can be difficult to track an application you wish to install. In fact, in our tests, when searching for some applications with Dash, we had to specifically choose the 'More suggestions' Scope. The applications were not suggested for installation otherwise, even when we typed the exact name as the query.

Thankfully, Dash still gives priority to the local search results, which means that installed applications and recently used files and multimedia stored on your hard disk are listed at the top of the search output. Results from Scopes are listed under their own distinct headings, such as Reference, Music or News, but this makes for a very cluttered appearance.

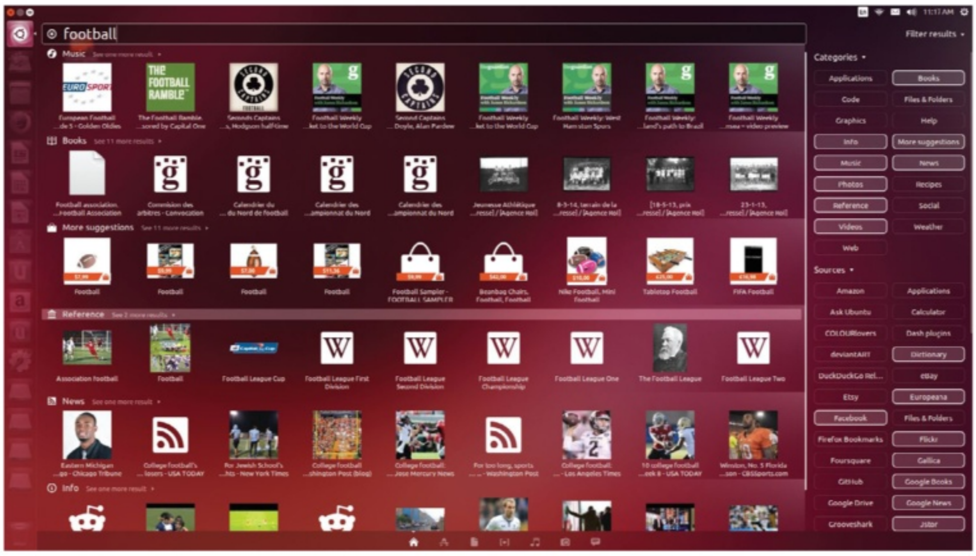
Clicking on an item in the search results presents more details about the item, along with an excerpt, if it's an Amazon product, a Wikipedia page or a news entry. You can then proceed on to the actual page if it turns out to be what you're looking for.

More mis-hits

Even though the latest release includes all these additional online sources for Dash, the distro hasn't done anything to address the privacy concerns voiced by users when Canonical decided to include Amazon search results in the previous 13.04 release.

While Canonical has said it doesn't store any identifiable user information, every time you use Dash, the query, your country, locale, search sources and the Scopes enabled on the system are transmitted to the company.

What's more, Ubuntu has even dropped the Recent Items tab in the rejig of the Security and Privacy controls. With that feature you could control application usage history. The only option that users get now is to turn off recording app activities entirely and not delete them selectively like you could earlier.



› You can't use the categories list on the right to jump to the relevant section in the results.

Additionally, as far as dropped plans go, Ubuntu was all set to adopt *Chromium* as the default browser for this release but for now *Firefox* still holds that particular position.

Mobilebuntu

On the face of it, Ubuntu 13.10 is the first release which includes a built-in experience for the mobile platform. In addition to the platform itself, it includes some mobile apps as well, such as a web browser, calendar, clock, weather and calculator.

Central to Canonical's plan for the mobile platform is its new and controversial *Mir* display server (*see Lie of the Land, below*). *Mir* is incorporated on the phone side of Ubuntu 13.10 but isn't on the desktop, despite it being marked as an important goal in the run up to this release.

Canonical's plan was to include *XMir*, a bridge between traditional X11 and the *Mir* display server, in 13.10, along with a fallback mode for certain hardware which would be phased out in 14.04. If everything went according to plan Canonical was hoping to complete the transition to *Mir* with Ubuntu 14.10.

Unfortunately, Canonical's plans have gone a little haywire, interestingly not because of any shortcomings in *Mir* itself, but rather *XMir*. Ubuntu developers have cited limitations with the *Xmir* multi-monitor support as one of the biggest unresolved issues.

Of the hundreds of Linux distributions to criticise and to celebrate, there's none that polarises the community like Ubuntu. Despite a very vocal community announcing Ubuntu's faults to anyone who'll listen, it remains one of the most popular Linux distro ever, with little risk of being displaced anytime soon (even if Mint continues to get more hits on its Distrowatch page than Ubuntu).

That said, Ubuntu 13.10 doesn't offer any compelling reasons to upgrade, especially considering the fact that Ubuntu 13.04 will still pump out security updates for a few more months. LXF

Lie of the land

A display server is a crucial part of the desktop. To replace the ageing X.org, Canonical is building *Mir* an entirely new display server. This is designed to take advantage of accelerated hardware and is part of Canonical's strategy of making sure Ubuntu works seamlessly across hardware devices. The developers also highlight *Mir*'s rich input handling that will make it easier to support things like touchscreen gestures.

Mir is already available on Ubuntu Touch devices, but the route to the desktop is a bit longer. To facilitate the transition to *Mir*, the developers are using *XMir* on the desktop. *XMir* is largely built on top of X.org and only uses *Mir* to paint the screen.

Due to this reason many developers fail to see the advantage of *Xmir*, which they believe is still hindered by the limitations of X.org. The delay of *Mir* in Ubuntu 13.10 has also attracted the wrath of developers upset by Canonical's decision to drop *Wayland*.

The *Wayland* project, created by AIGLX developer Kristian Høgsberg, aims to provide a display server protocol fit for composited desktops. The duplication of effort and the risk of fragmentation hasn't resonated well with the community, along with Canonical's claims that *Wayland* wouldn't meet its needs.

LINUX FORMAT Verdict

Ubuntu 13.10

Developer: Canonical
Web: www.ubuntu.com
Licence: GPL

Features	7/10
Performance	8/10
Ease of use	8/10
Documentation	9/10

» *A 'meh release, albeit the first with Ubuntu Touch included, that doesn't have much to get excited about.*

Rating 7/10

Sir, You Are Being Hunted (Alpha)

Graham Morrison gets lost in the crowdfunded game of stealth, menacing robots and constant updates.

In brief...

» Stealth and strategy abound in this survival game. Similar in tactics to *Thief.*

This isn't survival horror. If it were, the bad robots would look like zombies and there would just be night, and your task would be to find some way of escaping a dungeon hidden beneath an old Dutch Colonial styled house located in the midst of a forest. Instead, you find yourself dropped into the middle of a randomly generated pastoral landscape consisting of North, South East, West and Centre islands; an archipelago littered with stone circles and pieces of some kind of device, scattered after an experiment went wrong.

Luckily, whatever the experiment was, it heated up the pieces to such an extent that they now smoke, and it's your job to hunt them down by looking for rising smoke across the landscape.

This is where the robots come in. They have silly robot-like voices and often repeatedly say humorous things. Which is important, because if you can hear a robot, you're too close. Your best tactic is to avoid those robots, because if they see you, they start shooting.

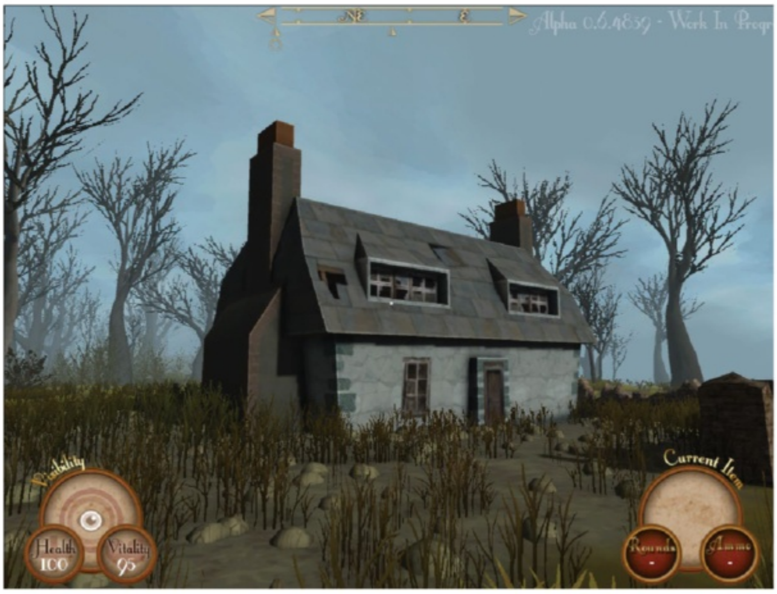
There are different classes of robots, and none of them are pleasant. If you're lucky enough to find a gun and some ammo, then you can take the robot hordes on, but this isn't advisable, as the noise usually brings only more robots; even ones hanging from hot air balloons, the red laser of their beady eyes visible from half a kilometre away. Far better to throw something in the opposite direction, sending the robots into the chase, while you somehow jump the fence, grab the smoking device fragment and run.

Food fighting

Food can be found, cooked and eaten to restore your energy, and this needs to be done with some regularity or you'll starve. Be careful if you cook over an open fire, though, as it's bound to attract company. Other objects can also be used, from bottles to sticks of dynamite, as well as some rags and medical supplies for tending to your inevitable injuries.

It's the robot AI that makes the game so much fun, and just so difficult. The robots take into account disturbed birds, for example, and they'll try and outflank any sounds they hear, often appearing to work together in thrashing you out of a copse of trees or a group of buildings. Sneaking up on a patrol, which you often have to do, has a real strategic element as you try to work out where they'll go next.

There's a lot of creeping along hedgerows to be done in autumnal desolation of the pastiche English landscape, and the robots add a sinister dystopian edge. Combine this with the limited ability to save, and you've an old-school tension that's rare in modern games. And despite its alpha state, we found it impressively compulsive. LXF



» Do be very, very quiet, Sir, you are being hunted... by robots.

Features at a glance

Robot AI

The AI in these top-hatted country clothed hunters makes the game challenging and fun.

Softy softly

Disturbing birds will alert pursuers to your whereabouts, and hiding is actively encouraged.

LINUX FORMAT Verdict

Sir, You Are Being Hunted

Developer: Big Robot
Web: www.big-robot.com
Price: £14.99

Gameplay	9/10
Graphics	7/10
Longevity	8/10
Value for money	7/10

» *A great fun and often tense experience that's getting better with each update.*

Rating 8/10

Ouya

David Hayward tries out the little games console that promised so much.

› **The Ouya looks great, but has some nasty flaws.**

In brief...

» Kickstarter's champion offers the potential for some star quality gaming in a small box, but is marred by some poor design decisions.

Specs

» **Features:** Tegra 3, quad-core CPU, HDMI, 1GB LPDDR2 RAM, USB, Wi-Fi and Ethernet.

And so begins the revolution. These are the first words that greet you on a transparent, orange perspex sheet when you open up the trendy, glossy and aesthetically designed packaging of the Ouya. The initial feeling is one of class, and that you've grabbed yourself a bargain in the form of the future of gaming consoles, but, unfortunately, you may be in for a let down.

The packaging of the Ouya is one thing, but the actual contents are something quite different. The Ouya really is a smart, Rubik's cube-sized device. The brushed aluminium and plastic case is wonderfully designed, and looks very attractive sitting next to the TV, and not out of place at all with the other mixture of modern appliances. Unfortunately, the Ouya's design is scuppered with last year's mid-range tablet technology. While this isn't too much of a problem for most users, the hard core among you will no doubt feel a little short changed, especially when comparing the specs with your current smartphone.

> “Ouya is scuppered with last year's mid-range tablet technology.”

There's enough connectivity to get you up and running: a HDMI port, Wi-Fi, Ethernet port, power, full-sized USB and a micro-USB port sit at the rear of the unit in a snug fashion that can get a little crowded once loaded up.

The power button is on the top of the unit, which emits a soft glow once pressed, and feels as sturdy as the rest of the unit.

› **The back of the unit can get a little crowded when its fully loaded up.**

Next up is the included controller which, despite the much advertised “from the ground up” modelling, looks like the strange love child of an Xbox and N64 controller. It's not uncomfortable to use, but there are some serious design flaws.

Oh-no

The four buttons, which spell out Ouya, feel below average and have the habit of sticking frequently. The triggers have far too much travel in them; the D-pad is too squishy, but the sticks are decent enough. The batteries for the wireless controller sit within the hand grips, which means you'll have to remove the upper panels to gain access. Normally this isn't too much of an issue, but in this case the plastic feels cheap and when being replaced never seems to fully click into place.

Powering up the Ouya for the first time involves the setup and account creation process – one that includes you having to add a valid debit or credit card, which in our mind is quite unforgiveable in this day and age. Refusal to enter such information means you'll be returning the unit to the shop, as you won't be allowed to continue any further.

Once in though you're greeted with a customised Android 4.3 UI, which isn't too bad. The Discover Ouya store has a decent selection of games, with a few apps, such as VLC and XBMC, all arranged in tiles and groups. The tiles themselves offer very little detail, and despite every game being free to try (or free to own), you'll have to enter the game and initiate the purchase feature to see how much you'll have to spend. It's far too long-winded to be bothered with most of the time.

The games in the store are very basic. Some are great fun, but hardly stretch the capabilities of the technology; others are very homemade, and although good, don't necessarily warrant its £100 cost. Retro emulation is the biggest draw at present, but one that's sure to be short lived.

It's not that the Ouya is diabolically bad; it's just that it doesn't deliver what was envisaged. Perhaps after further development it may have more content to attract the modern gamer. But for now, with the next gen consoles looming on the horizon, you have to ask yourself, is it really worth it?

In many ways the Ouya has become a poster child for what can be achieved through the power of the masses. It may very well fall short of the mark, but in terms of real potential, then it could be brilliant – it's just not there yet.

Verdict

Ouya

Developer: Ouya
Web: www.ouya.tv
Price: £99

Features	6/10
Performance	7/10
Ease of use	6/10
Value for money	6/10

» *Heaps of potential, but a poor controller build and bland content significantly mar what Ouya offers.*

Rating 6/10

PiGlow

Is **Les Pounder** dazzled by the latest add-on board from Pimoroni?

In brief...

» A Raspberry Pi add-on board that provides 18 controllable LEDs. Also consider: Pi Lite or Pi Matrix.

Pimoroni, the Sheffield based Raspberry Pi accessories business, has enjoyed a year in the limelight with its PiBow case being the popular choice for many Raspberry Pi owners, and much media coverage of it PiBow toxic case. But Pimoroni don't just make cool cases, it has also successfully funded a Kickstarter campaign to create a Raspberry Pi power mini-arcade cabinet, called the PiCade. And just recently it launched its latest product, the PiGlow.

PiGlow is an add-on board that fits neatly over the GPIO (General Purpose, Input Output) pins of your Raspberry Pi. It has eighteen exceptionally bright LEDs: three each of red, orange, yellow, green, blue and white. PiGlow is extremely easy to fit; you only need simply push the device onto the pins. As you might expect, PiGlow has been designed to be compatible with Pimoroni's PiBow case. The PiGlow might fit over all of the GPIO pins, but it actually only uses six of the pins (Pins 1, 2, 3, 5, 14 and 17 if you are interested), which means you can feasibly break out the PiGlow to a breadboard and still be able to use the rest of the GPIO pins.

PiGlow uses the SN3218 8-bit 18-channel PWM chip to drive 18 surface mount LEDs and it communicates with the Raspberry Pi via I2C (Inter-Integrated Circuit) over the GPIO header. Each LED can be set to a PWM (Pulse Width Modulation) value of between 0 and 255, which means that you can create gradual fades to the LEDs, allowing you to dim their brightness.

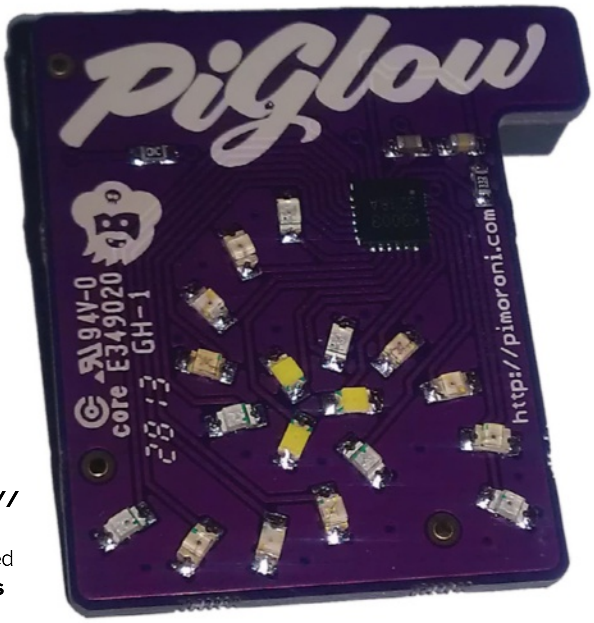
Installation is quite easy and only differs based on how you would like to interact with PiGlow. There are many ways to interact with PiGlow, Python, Scratch, WiringPi and Node.js.

Get it glowing

For our review, we used Jason Barnett's excellent Python classes, which you can download from **https://github.com/Boeeerb/PiGlow**. Installation required us to install **python-smbus** and **python-psutil**, which allows us to use the I2C bus on the Pi. Once they were installed, we edited **/etc/modules**, ensuring that **i2c-dev** and **i2c-bcm2708** were at the end of the file. The last bit of post installation configuration, was to edit **/etc/modprobe.d/raspi-blacklist.conf**, ensuring that **blacklist spi-bcm2708** and **blacklist i2c-bcm2708** were commented out. We then rebooted the Raspberry Pi, and made a new folder in our home directory called **piglow**, from there we downloaded two Python scripts, **piglow.py**, which contains the Python code necessary for PiGlow to work, and **test.py**, which tests the PiGlow by asking you to set the brightness of each group of coloured LEDs.

If you download all of Jason's Python classes, you can see the broad scope of what this device offers, for example the CPU script changes the colour of the LEDs, and depending on how hard your Raspberry Pi is working, is really awesome to see.

You might be wondering what you can do with PiGlow? Well, whatever sparks your interest in the way of lighting. PiGlow does certainly provide you with a novel method to add interesting output to our projects.

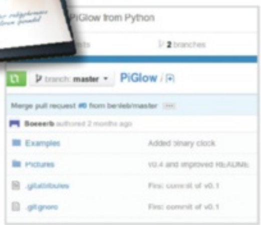


» If the PiGlow isn't enough you can always add glowing USB cables, too.

It enables new project ideas, ranging from mood lighting, to showing the current system load and because PiGlow has such a versatile library of languages that can be used with it, you can easily incorporate it into your Raspberry Pi projects. The only downside that we found, is that it blocks off the GPIO port, meaning that you cannot easily incorporate the PiGlow into an existing project that uses the GPIO, even if in practice the PiGlow only needs a few pins, but as we mentioned earlier, you can break the PiGlow out to a breadboard.

This is a great piece of kit, though, and at such a cheap price, well worth the money to use in your projects. LXF

Features at a glance


PiGlow from Python
2 branches
branch: master PiGlow /
Merge pull request #6 from benleb/master
Boeeerb authored 2 months ago
Examples Added binary clock
Pictures v0.4 and improved README
.gitattributes First commit of v0.1
.gitignore First commit of v0.1

Languages

PiGlow is supported by Python, Scratch, WiringPi and even NodeJs, so it can be used in any project.

GPIO Pins

PiGlow connects to all the GPIO pins, but only needs a few, so you can use PiGlow with a breadboard.

LINUX FORMAT Verdict

Verdict Box

Developer: Pimoroni
Web: http://shop.pimoroni.com/
Price: £9

Features	8/10
Performance	9/10
Ease of use	8/10
Documentation	8/10

» *A novel piece of kit that provides great output in a unique way. Add it to your Raspberry Pi project now.*

Rating 8/10

Cloudonomics

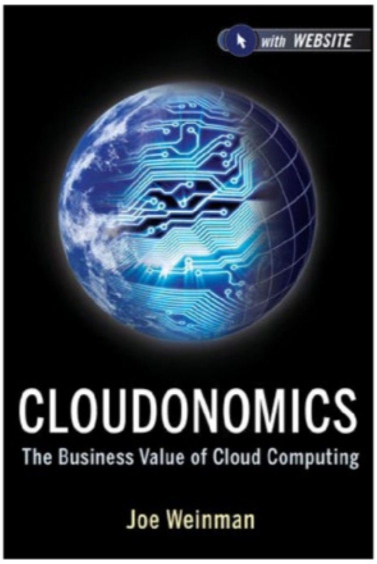
Andrew Gregory always liked cloud computing – now he knows why.

We're used to publishers making up portmanteau words grafting an established idea onto a buzzword in an effort to drum up sales. It was, therefore, with great trepidation that we picked up *Cloudonomics*, fearing a re-hash of basic microeconomic theory with a few soon-to-be outdated examples of how Yahoo, Microsoft or Google were doing things in 2010. We were pleasantly surprised.

This book takes what educated computer users understand about the cloud and distributed computing models (it's more scalable and flexible, and therefore cheaper/more efficient than a centralised solution) and backs that general point up with masses of theory. Each chapter has a list of references, and there's a decent index, which reveals the book's ambitions as a serious textbook rather than something to be filed under popular science.

There are also plenty of real-world examples to back things up. Don't expect Tim Harford-style vignettes to illustrate the issues here – the subject matter is far too specific to be summed up in bite-size parables. However, it's accessible enough that even if you don't understand the sigmas and deltas, you'll grasp the content.

This isn't a beginner's guide; there's some heavy-duty maths here. But if you prefer to have someone else do your calculations for you there's always **www.cloudonomics.com**, where you'll find some of the concepts explained in the book, modelled before your very eyes. Excellent stuff. LXF



› More a business than an economics book, there's nothing nebulous about the ideas put forward here.

LINUX FORMAT Verdict

Cloudonomics

Author: Joe Weinman
Publisher: Wiley
ISBN: 978-1-118-22996-5
Price: £33.99
Pages: 391

› *Several cuts above the usual vague cloud/business waffle.*

Rating 8/10

Learning Python (5th edition)

Graham Morrison discovers that his beloved C++ might have met its match.

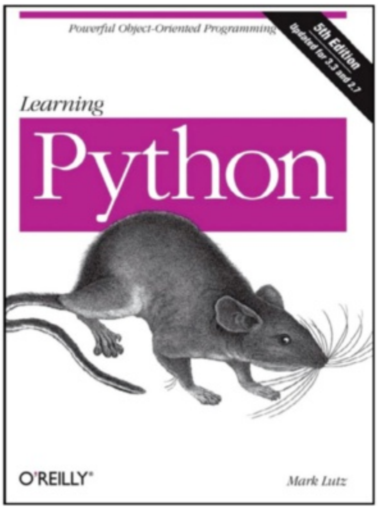
LINUX FORMAT ★★★★★ TOP STUFF

At over 1,500 pages, this book is 50% bigger than *The C++ Programming Language* by Bjarne Stroustrup. Which is quite an accomplishment, and not too alien a comparison. Python's creator, Guido van Rossum, admits that C was one of the big influences on his language. But he could never have envisioned how widely used his language would become, and how large a book on starting Python would need to be.

Fortunately, this isn't just any book. This isn't a book designed to entice non-programmers into the world of programming, nor to show the great unwashed how easy programming is. It's a guide to everything that you need to know about Python from a programmer's perspective.

You don't need to be an expert to get the most out of it, but you should know some first principles. The word we're looking for is 'comprehensive', but that implies a reference book, and this isn't one of those either. It's a book for those wanting to learn Python seriously, and will turn even occasional tinkerers into developers who could justifiably put Python down on their résumé.

As this is the fifth edition, and because it's based on a training course, many of the speed bumps have been ironed out. What's great about this edition is that it no longer assumes that the 3.x version of Python will supplant 2.x, preferring to refer to them as either a prototyping language or a stable 'workable' platform. All the examples provided are now given in both dialects, turning the book into a great comparison if you want to understand the advantages of either. It's brilliant, time-consuming, well written, and yes – comprehensive. LXF



› If you want to take your Python skills further, look no, er, further than this.

LINUX FORMAT Verdict

Learning Python

Author: Mark Lutz
Publisher: O'Reilly
ISBN: 1-44935-573-0
Price: £49.99
Pages: 1,540

› *If you have even half an inkling of getting into Python, this is the book to put on your Christmas list.*

Rating 10/10

Roundup

» Every month we compare tons of stuff so you don't have to!

Beginner distros

Ready to take the Linux plunge but unsure of which distro to go with? **Shashank Sharma** lists the five best choices for newcomers.

How we tested...

All distros were tested on the same dual-core machine with 4GB RAM. We've selected the latest stable releases for all the distros, except for SolusOS. The distro has made significant changes since its last stable release, so we've settled on an alpha release for the roundup.

For inexperienced users, the documentation is one of the most important reasons for choosing a distro. The distribution also needs to be easy to install. Since most users of these distros have probably never installed Linux before, this is a very important feature. Just as important is software management and the kind of apps that are included in the distro.

Apart from these, the distro also needs to be easy to use for day-to-day activities. The ideal distro for newbies is one that does all of the above and also makes it easy for them to tweak some settings.

Abundance of choice is one of the biggest challenges faced by new Linux users. Choosing your first Linux can be a very daunting task. Especially when you don't even know what criteria to look for when deciding on a distro.

In the mid-to-late 90s, choosing a distro was a much simpler process. You went with the distro you had heard about, or the one that someone you knew had experience with, or the one with some degree of documentation. Naturally, then, you were limited in choice to RedHat, Debian, or Slackware. While those criteria still apply, the sheer number of Linux distros available now, and their vocal fan bases, makes it difficult to settle on one and get started.

We've deliberately shied away from the popular mainstream distros, as we didn't just want easy-to-use distros. Instead, we've selected five that we believe are ideal beginning points.

Ubuntu has long been a popular Linux distribution, but it isn't quite right for beginners. However, it can be with the right changes. This is why four distros in our list are Ubuntu-based.

For users hoping for familiarity as they move away from a proprietary OS, we've got a distro each that resemble Mac and Windows.

> "For this roundup, we've deliberately shied away from the popular mainstream distros"

Our selection

- » Zorin OS
- » Pinguy OS
- » Elementary OS
- » SolusOS
- » Pear Linux

Included software

What does it offer out of the box?

Distributions are usually designed with the need to serve the most possible users in mind. This philosophy also drives the applications that are bundled with them. All the distros in our list offer the minimum, such as internet browser, email client, text editor and media player. But if you expect lots more apps, they have those as well!

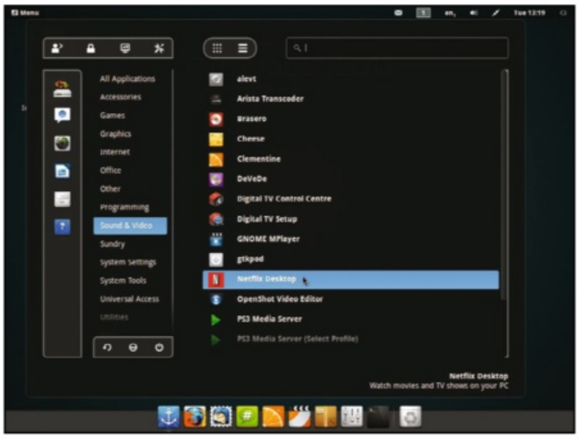
The current release of SolusOS is meant for developers and testers, and has a limited number of apps. Of note are *Firefox 24.0b9* and *Thunderbird 17.0.8*. The developer has already announced plans of shipping the Steam client in the final release, and you can expect all sorts of productivity and multimedia apps.

Zorin is bristling with apps. You get the usual office and internet apps, such as *LibreOffice* and the proprietary *Google Chrome*. The distro also lets you view content in proprietary formats from within the live environment. Also included is *Gimp* image editor, *Shotwell* photo manager, *Thunderbird*, *Pidgin* for IM, *Totem* video player, *Rhythmbox* music player, and the *OpenShot* video editor. It also includes *Wine* and *PlayOnLinux* to install Windows-only apps and games. There's also *Web Browser Manager*, which makes it easy to install different browsers. The distro offers *Gwibber*, a desktop app that lets you control most of the popular social networks, such as Twitter, Facebook, Google+ and Flickr.

Elementary OS provides a simple, elegant design. It has apps with a simple design. This is explained by the inclusion of *Geary* email client and *Shotwell* photo manager. Most other distros ship with *Thunderbird*, although *Geary* is a smart email client. You also get *Totem* movie player, *Noise* music player, and *Midori* web browser. The latter two reaffirm the distro's fondness for lightweight, simple apps. Elementary provides fewer default packages and you need its software management app to install the ones you want.

Pear OS ships with *LibreOffice*, *Thunderbird*, *Firefox 20*, *VLC* media player and *On Air* music player. It also offers *Time Back*, a clone of Apple's *Time Machine* backup tool, *Empathy* for IM and *Cheese* webcam tool.

PinguyOS ships with *Thunderbird*, *LibreOffice*, *Empathy*, *Deluge*, *TeamViewer 7*, *DeVeDe* to burn discs, *OpenShot* video editor, *Cheese*, *Clementine*, *PlayonLinux*, *Gparted 0.16.1*, *Shutter*, *PlayOnLinux* and more. The inclusion of *TeamViewer* seems like a masterstroke, as the app allows you to control remote desktops.



› Hovering the mouse over applications reveals what each of them does.

Verdict

Zorin OS
★★★★★

PinguyOS
★★★★★

Elementary OS
★★★☆☆

SolusOS
★★☆☆☆

Pear Linux
★★★☆☆

» Pinguy offers some unorthodox but brilliant packages.

Software management

For when you need to install additional applications.

For most new users, the default set of apps should be more than enough to get started. As you become more accustomed to your distro, you may wish to install additional apps. Software repositories may seem like a strange concept at first, but most distros provide useful tools to help you install software easily.

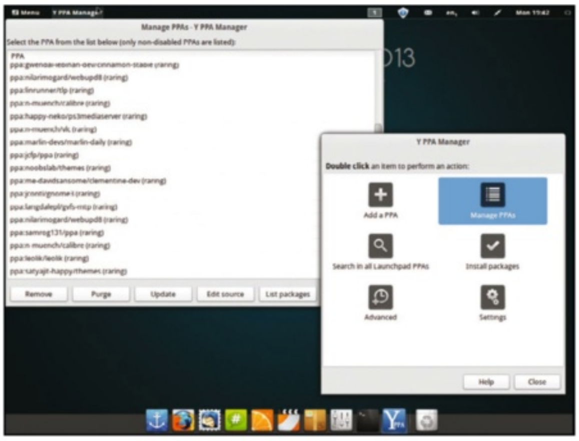
SolusOS uses *gpk-application* as a frontend for installing apps, but the developer is working on a custom GUI for its *PiSi* package management system.

Elementary OS is working on its own AppCenter as a replacement for *Ubuntu Software Center*. With the exception of Elementary OS, which ships only with *Ubuntu Software Center*, all other Ubuntu-based distros in our list ship with that and *Synaptic Package Manager*.

In Pear Linux, neither of these tools is accessible from the desktop. You must first head into Launcher and then click the System Tools tab. Main, restricted, and multiverse repositories are enabled by default.

With Zorin, you get a redesigned *Ubuntu Software Center*, as well as the *Synaptic Package Manager*. In addition to all the usual repositories, it also has repos for Google and *Opera*.

PinguyOS has a variety of additional software repositories that are enabled by default. It also includes repos for Linux Mint, Ubuntu and Elementary OS. There are PPAs for themes, and apps such as *Clementine*. The distro ships with too many PPAs, but thankfully it also includes *Y PPA Manager*. This is a tool that you can use to make sense of and manage all the PPAs.



Select the PPA from the list below (only non-disabled PPAs are listed):

PPA
ppa:gwendal-lebihan-dev/cinnamon-stable (raring)
ppa:nilarimogard/webupd8 (raring)
ppa:linrunner/tlp (raring)
ppa:n-muench/calibre (raring)
ppa:happy-neko/ps3mediaserver (raring)
ppa:n-muench/vlc (raring)
ppa:marlin-devs/marlin-daily (raring)
ppa:jcfp/ppa (raring)
ppa:noobslab/themes (raring)
ppa:me-davidsansome/clementine-dev (raring)
ppa:jconti/gnome3 (raring)
ppa:langdalepl/gvfs-mtp (raring)
ppa:nilarimogard/webupd8 (raring)
ppa:samrog131/ppa (raring)
ppa:n-muench/calibre (raring)
ppa:leolik/leolik (raring)
ppa:satyajit-happy/themes (raring)

Double click an item to perform an action:

› That's a lot of PPAs, yo!

Verdict

Zorin OS
★★★★★

PinguyOS
★★★★★

Elementary OS
★★★★☆

SolusOS
★★☆☆☆

Pear Linux
★★★★★

» SolusOS loses out, as package management is a core skill.

Usability

How easy is it for newbies to configure the desktop?

You can tell a user has found a distro that they like when they begin to tweak its different aspects. This moving away from the defaults is a sign of maturity for any user, but especially so with new Linux users.

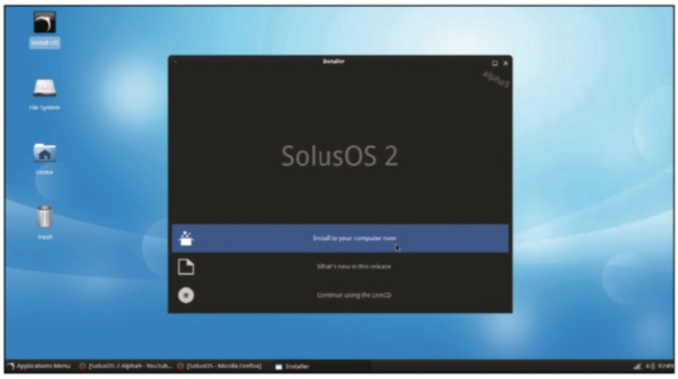
People often say Linux distros are extremely customisable. But what does it mean for new users? Sure, you can change the desktop background, the icons theme, define keyboard shortcuts, configure power management and make other changes to the appearance and behaviour of the distro. But is it easy for a first-time Linux user to do all of that?

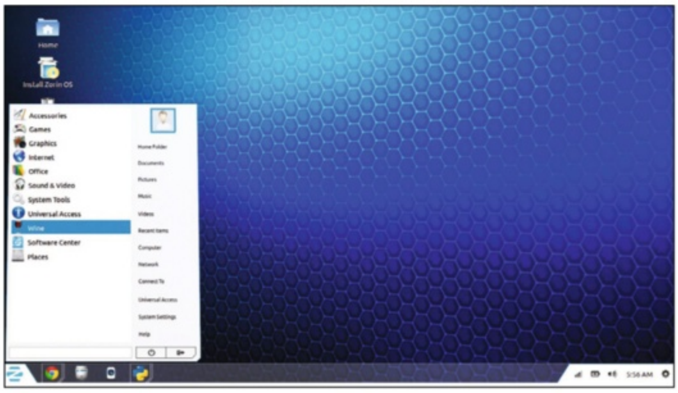
While all the distros in our list allow you to do all of this and more, they each go about the process differently. If the distro is aimed at new users, it earns high points if it includes special custom tools to help the users easily customise the distro to their liking.

SolusOS ★★★☆☆

While originally based on Debian, SolusOS is now being developed from scratch by the developer who worked on the Debian edition of Linux Mint. Although the distro is under active development and hasn't even had a beta release yet, it is pretty stable.

The Alpha 9 release has a basic installer and even lacks a partitioner, but the developer is adapting the partitioning tool from Pardus's *YALI* installer. The final SolusOS 2 will ship only non-live installable images. The distro uses its own adaptation of the *PiSi* package manager, created for Pardus Linux. It supports delta upgrades, which lets users update their system using minimal bandwidth. It uses *Xfce*, but the completed distro will ship with *Consort*, its own fork of the Gnome Classic desktop.


SolusOS 2
Install to your computer now
What's new in this release
Continue using the LiveCD



Zorin OS ★★★★★

This is one of the finest distros to attract inexperienced Linux users. It has everything to offer a nice usable experience to users coming from another Linux distro or even from Windows or Mac OS X.

Besides its Windows 7-styled desktop, the custom application launcher does a pretty good job of mimicking the Windows 7 Start menu. The Core edition has enough to whet your appetite, and you can shell out some money to get the specialised versions.

Zorin also includes all of the Ubuntu goodies, such as *Ubuntu One*, which is well integrated into the distro. It also instills good desktop practices by regularly reminding users to set up the backup app. All in all, the distro has the right mix of the best of Ubuntu sprinkled with some custom Zorin apps, such as the *Look Changer*.

Commercial services

What paid add-ons does the distro offer?

A distro can have several reasons for offering paid add-ons. More often than not, it's just the developers trying to make some money so they can continue to produce it. This is why some distros also enable users to make donations to the project.

In addition to the desktop release, Zorin OS produces four premium versions that can be downloaded after giving a donation. The Business, Multimedia and Gaming versions can be had for a minimum of €7.99, while the Ultimate edition can be downloaded after donating a minimum of €9.99. With a purchase of these editions, you also get premium support. The distro also sells merchandise, such as mugs, t-shirts and stickers on CafePress.

PinguyOS also has an extensive store on CafePress, from where you can get all kinds of merchandise, such as mugs, t-shirts, bags and baby bibs. You can also donate via PayPal.

To show your support for Pear Linux, you can get stickers. You can also donate to the project; there are several options, ranging from the minimum of €5 to the maximum of €100.

You can also fund the developers' salaries by becoming a sponsor. There are five levels of sponsorship: Platinum, Gold, Silver, Bronze and Community. Sponsors pay a fixed amount per month, and get exposure on the website and monthly announcements. The Elementary store lets you buy the 32-bit CD for the current release and also offers a few t-shirts and stickers.

Verdict

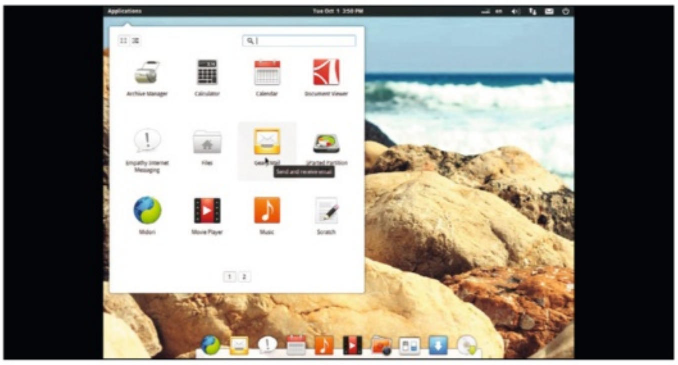
Zorin OS	★★★★★
PinguyOS	★★★★★
Elementary OS	★★★☆☆
SolusOS	☆☆☆☆☆
Pear Linux	★★★★★

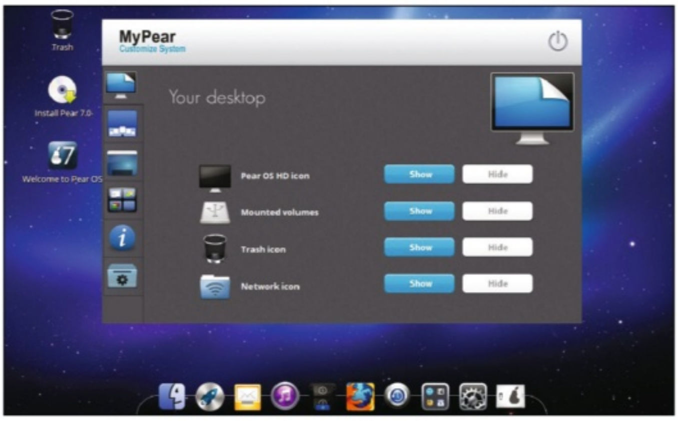
» *SolusOS has no commercial services, but it's not a poor distro.*

Elementary OS ★★★★☆

This is one of the simplest Ubuntu-based distros available, and is a good starting point for beginners. The distro places great emphasis on design, and this has resulted in a curious choice of default software packages. While these may not be to everyone's liking, the apps are highly usable and a suitable replacement for their more popular alternatives.

The distro is lightweight and blazingly fast. It doesn't offer many apps out of the box and doesn't include codecs for proprietary media formats. This means that you can't play MP3s, videos, or even YouTube videos out of the box, but you can leverage its Ubuntu lineage and access thousands of additional packages and multimedia codecs using the software centre.





Pear Linux ★★★★★

This distro strives to be Mac-like in its appearance. Like Elementary, it too doesn't offer many apps out of the box, which is rare for Ubuntu-based distros. It's only available as 64-bit ISOs for now, but future releases will also provide a 32-bit variant, starting with Pear OS 8.

Aside from the Desktop edition, the team also produces a Server edition, which ships with *Apache*, *MySQL*, *Samba*, *Webmin*, *TomCat*, etc, and can power your business, home office or school. It also includes a custom app called *MyServer* to help you manage the different services.

The desktop edition includes *MyPear*, a one-stop tool to control and configure many different aspects of the desktop, such as defining the hot corners, the positioning of the panel, window animations and more. There's also *Clean My Pear* to help you maintain your system. You can use the tool to empty trash and delete temp files and browser cache.

PinguyOS ★★★★★

Traditionally, any application or operating system with Beta in its name implies the project is not ready for the masses, but Pinguy has turned that theory on its head, and it's out-every-six-months final releases are all Beta.

The distro is wonderfully stable and a very attractive option for all Linux users. Whether you're an absolute beginner or someone looking to switch to another distro, this is definitely worth your time.

It ships with a custom *Docky*, which you can use to create a number of docks. To each such dock, you can add docklets, such as weather, a network usage monitor and a workspace switcher. The distro also includes the *Tweak* tool to help you easily configure many different aspects of the desktop.

Support and documentation

RTFM doesn't help. What else have you got?

Regardless of a user's past OS dalliances, a beginner in Linux will encounter a vastly different way of doing things; everything from appearance to the alternative apps they will need to master.

This is why the distro must provide extensive documentation. Additional resources, such as forum boards, mailing lists, wikis, etc, which can help a newbie tap the collective experience of the community, are also appreciated.

Elementary OS provides to-the-point, easy-to-understand documentation on the website. The project also has an Answers page, where anyone can post a question. It also lets you connect to the project's IRC channel from within the browser.

SolusOS has a neatly arranged forum board, which hosts community contributed tutorials and tips and tricks.

While it provides only a bare bones installation guide, Zorin OS more than makes up for it with its many FAQs for new users, forum boards with installation help, howto's and tutorials. The project also has an IRC channel, so you can have your questions answered instantaneously.

Except for the FAQs, PinguyOS offers its users everything that Zorin does. What's more, there's also a very thorough step-by-step installation guide to help you out.

In contrast to these distros, Pear Linux provides only a forum board and little else. While the website also lists a wiki, it is inaccessible for now.

Verdict

Zorin OS ★★★★★

PinguyOS ★★★★★

Elementary OS ★★★★☆

SolusOS ★★★☆☆

Pear Linux ★★☆☆☆

» *Documentation may be one of the most important features.*

Release cycle

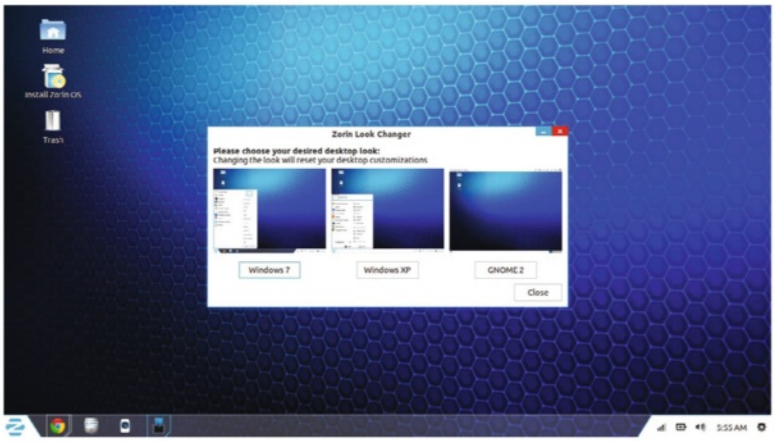
Which schedule is best for a beginner?

There are three popular development methodologies that Linux distros typically adhere to – fixed schedule, fixed feature and rolling release. With a rolling release comes a learning curve that may be too steep for most new users. It's because of this reason that distros such as Gentoo and Arch are not recommended to newbies. With a fixed feature schedule, the distro is released when it's good and ready – there is no fixed date for a release. SolusOS is the only distro on our list that follows this practice. It's being built entirely from scratch, which is why it has no 'upstream source'. Since the distro follows the fixed feature strategy, the date for the final release is not fixed. The fixed schedule is one of the most popular release cycles, and is followed by the majority of distros. In the fixed schedule, a new release is pushed out at fixed intervals, usually every six months. Ubuntu follows this twice-yearly release cycle and so, naturally, its derivatives do the same.

Zorin OS is based on the latest Ubuntu release. Work on a new edition begins as soon as a new Ubuntu release arrives on the horizon, but it takes time for the developer to produce the different editions.

The current Elementary OS release is based on Ubuntu 12.04 LTS. It will not, however, produce a distro based on Ubuntu 13.10. The next release of Elementary, named Isis, will use Ubuntu 14.04 as its base.

Pinguy's six-month-releases ship with bleeding-edge software, and are not considered stable. They remain in beta, despite being a final release. The stable releases are based on Ubuntu LTS releases.


Zorin Look Changer

Please choose your desired desktop look:

Changing the look will reset your desktop customizations

› **Zorin ships with several custom tools, such as the *Look Changer*, which offers a choice of Windows 7, XP or Gnome 2 on the freely-downloaded edition.**

Verdict

Zorin OS
★★★☆☆

PinguyOS
★★★★☆

Elementary OS
★★★★★

SolusOS
★★★☆☆

Pear Linux
★★★★☆

» *The distros may be based on Ubuntu, but look nothing like it.*

Installation

How user-friendly is the process?

The Live CD is the best thing Linux has to offer to new users. The ability to test a distribution and familiarise yourself with its ins and outs without installing it helps ease new users into Linux.

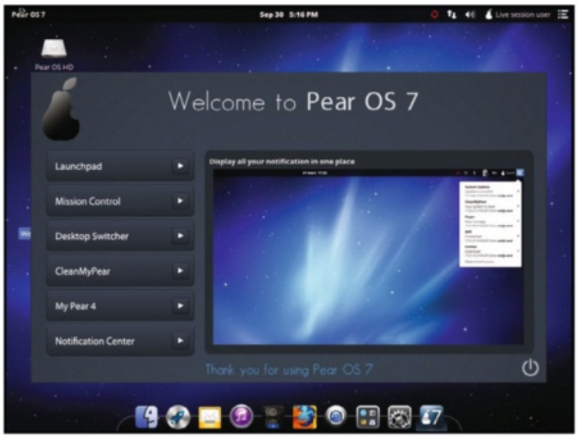
All the distros here let you test them from the Live environment. After using the distro if you feel its right for you, you can run the installation from within the Live environment. Most of these distros have an icon on the desktop you can double-click to launch the installation.

As a new user, the installation needs to be easy. It's likely that a user already has some form of an operating system on the machine. If that's the case, the user will have to partition and resize the hard disk. This is the step where many distros aimed at new users falter. But it's not just a problem for newbie-centric distros. Many mainstream distros fare poorly because they don't provide a friendly-enough installer.

Like almost every aspect of SolusOS 2, its installer is under active development. The current installer is based on the one the developer wrote while working on LMDE and later used in SolusOS 1. In its current state, the installer is very basic – it doesn't even have a partitioner. This means users will have to rely on other partitioning tools to prepare their hard disk for the distro. The developer is working to adapt the partitioning tool from Pardus's *YALI*.

The Ubuntu-based distros all use Ubuntu's *Ubiquity* installer, somewhat modified, to better suit the beginners the distros aim to target.

The installation process usually takes you through seven or so steps that cover partitioning, creating a user, defining the time zone and specifying the keyboard layout. The most important step is partitioning, where you can erase the entire disk and use it to install the distro, or specify a custom partitioning layout. The best thing about using *Ubiquity*, as a newbie, is there's plenty of documentation. Plus there are YouTube videos that take you through the installation process for each of our Ubuntu-based distros.


Welcome to Pear OS 7

Display all your notification in one place

Thank you for using Pear OS 7

› **The Ubuntu derivatives all have a similar installation.**

Since the distros are based on Ubuntu, you don't get to choose the software that is installed. Once you specify the disk and configure the partitioning, the distro will automatically install all software for you.

Verdict

Zorin OS
★★★★★

PinguyOS
★★★★★

Elementary OS
★★★★★

SolusOS
★☆☆☆☆

Pear Linux
★★★★★

» *Ubiquity scores highly. It is unrivalled in its simplicity.*

Distros for new users

The verdict

The Linux ecosystem is often praised, and sometimes criticised, for giving users too much choice. This is true, not just for applications, but also for distributions. Untill a few years ago, it was considered the height of cool for experienced Linux users to complain about this distro proliferation, but nothing much came of that. Eventually, people turned to writing about how we will soon witness the year of Linux on the desktop. People still sometimes talk about that.

But there are some of us who are thrilled by each new Linux distro announcement. As new Linux users, you might get vertigo browsing through the list of distros, but this isn't a bad thing. It means that there definitely is a distro that's just right for you.

But you can't wait for this tailored-for-you distro to fall into your lap before starting with Linux. So what do you start with? How about one of our designed-for-newbie distros?

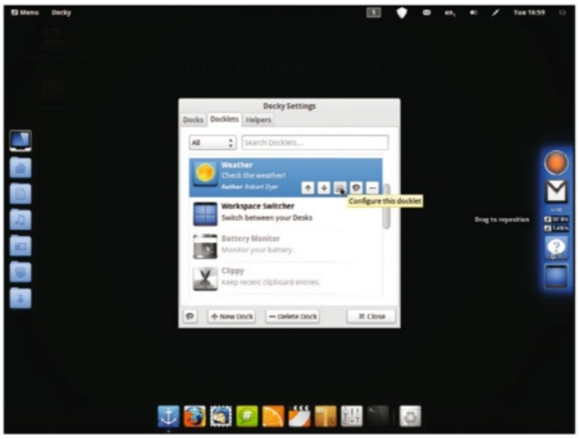
SolusOS appears to be the worst in the list, judging by the ratings. But this is a promising distro that we urge everyone to come back to once the final version is released. We've pitted an alpha release against more mature, stable distros, but the project still stands out. SolusOS 2 differs greatly from the last stable release, and the project has some wonderful ideas to attract and retain users.

Pear Linux is ideal for those looking to move away from Mac, but who still want some degree of familiarity. It's not as polished as Elementary, but is being actively worked on, and its custom set of tools make it stand out.

Elementary OS started off as a contender for the top slot, but small nags such as a reduced selection of default apps mean it finishes in third place. It faces stiff competition from Pear Linux, and may soon be overtaken by the French distro.

It was a close contest between Zorin and Pinguy. Zorin produces several commercial variants and includes custom tools.

For novice Linux users, Pinguy provides the best desktop experience. It is easy to use and configure, and has an intriguing design.



Docks and docklets can make the desktop attractive, if you think them through.

> **"Pinguy provides the best desktop experience. It's easy to use and has an intriguing design"**

1st PinguyOS ★★★★★

Web www.pinguyos.com Licence GPL and others Version 12.04

A pleasant-to-use distro. Perfect for newbies.

2nd Zorin OS ★★★★★

Web www.zorin-os.com Licence GPL and others Version Zorin OS 7

Very thoughtful distro. Good for most new users.

3rd Elementary OS ★★★★☆

Web www.elementaryos.org Licence GPL and others Version Luna

Nowhere near Pinguy or Zorin, but very usable.

4th Pear Linux ★★★★☆

Web http://pearlinux.fr Licence GPL and others Version Pear Os 7

A little bit of effort could well put it on the podium.

5th SolusOS 3 ★★★☆☆

Web www.solusos.com Licence GPL and others Version 3

Wait for the final release before you write it off.

Over to you...

Are you a new Linux user who has already tried out some distros? Let us know your favourites at **lxf.letters@futurenet.co.uk**

Also consider...

We chose not to go with any of the mainstream distros. There are those who believe there is no such thing as a newbie-centric distro, and that a determined user will find a way to persevere with a distro no matter how alien it feels.

We've often seen Arch and Gentoo recommended to new users, along with the all-time favourites Debian, Slackware, Fedora, Mint, Ubuntu, and so on. Gentoo and Arch will teach you Linux internals like no other distro could. But experienced Linux users still shy away from them, and with good reason. These two are not for those who've never used Linux before.

Depending on your past computing experience, you may find the five distros in our list too simplistic. In which case you can pick Fedora, Slackware, Debian, or any other.

If you are an absolute beginner, you'd be better off starting with one of our five choices, before switching to one of the others after a while. LXF

Subscribe today

Subscribe today and get Linux Format for just £3.85 an issue

Take advantage of our fantastic subscription offer and subscribe to *Linux Format* for just £25.49 every 6 months.

Want the digital edition? Turn to page 66!

Our subscribers are our most loyal readers, and as a thank you we like to make sure you not only get a great price, but also tons of other benefits.

Get all this:

- Exclusive access to the LXF subs-only area, featuring magazine PDFs, complete issues and coverdisc downloads.
- A digital edition of every issue from our subs-only area on your PC, iPhone or iPad.
- Save money on the shop price.
- Get each issue delivered to your door – a week before it hits the shops (UK only).
- Discount on *Linux Format* merchandise.

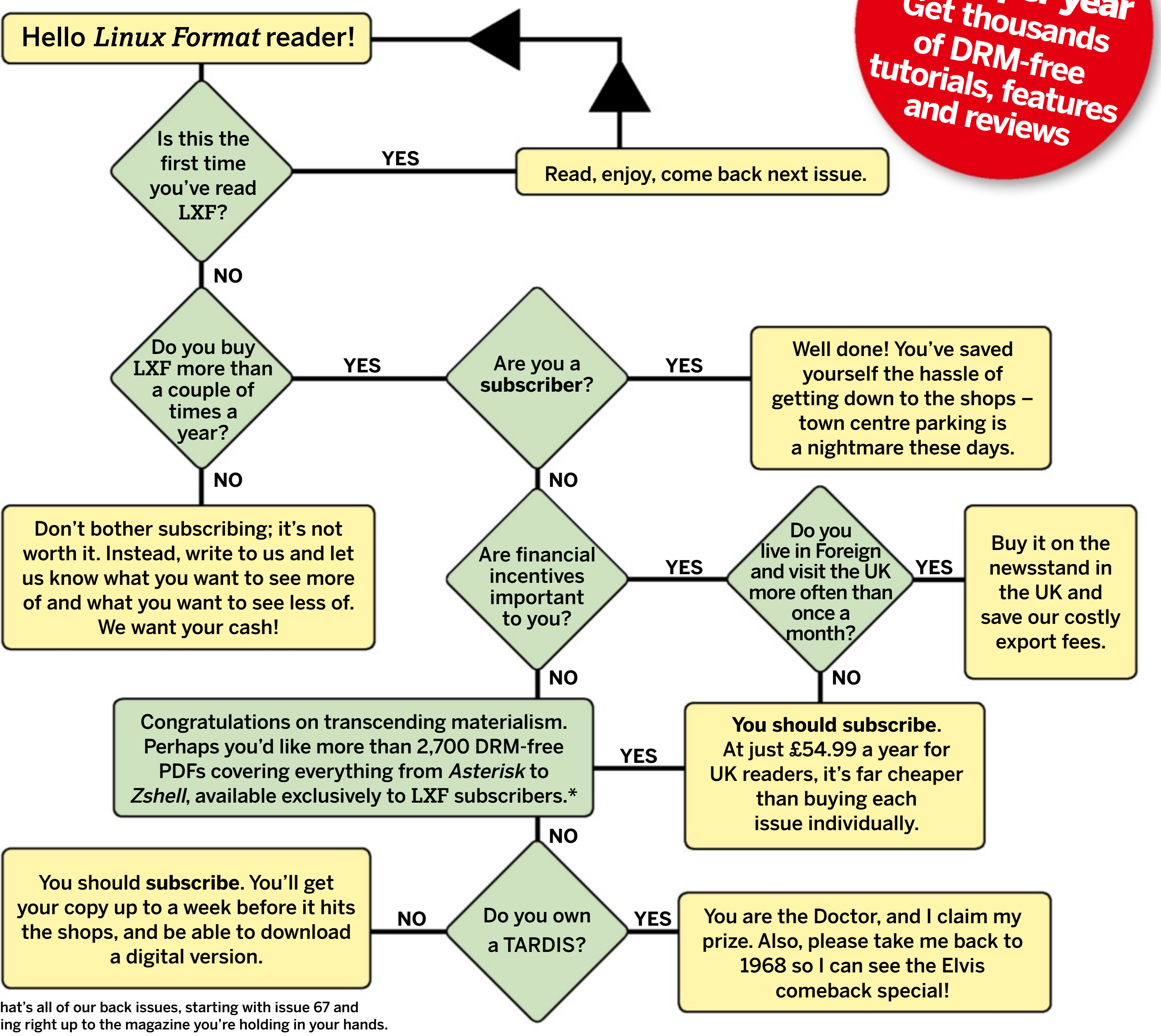
What You'll Pay:

- UK – Pay £25.49 every 6 months or £99.99 for 2 years
- Europe – Pay £99.99 for a years subscription
- Rest of the world – Pay £107.99 for a years subscription

Roundup: Best beginners' distro
LINUX FORMAT
#1 for Free Software
ON YOUR FREE DVD Elementary OS Luna, Hotpicks, Pinguy OS 13.04 64-bit + more!
Fund your dreams
How crowdfunding is transforming the economy of free software
» Get the backing you need
» Synergise the interwebs
» Become your own boss!
6 pages of programming excellence
CODING ACADEMY
Fortran Hack like it's 1957 p88
Python Print hello world p84
Inside Qt
» How one tiny graphical toolkit took the world of mobile development by storm p46
IS YOUR FREE DVD MISSING? Speak to your newsagent now!
Renaissance man
Most people today write more than Charles Dickens did in his lifetime
Robert 'r0ml' Lefkowitz on, well, everything p42
Also inside...
Ubuntu 13.10
Has Ubuntu lost it/still got it/got it back/always been ace? p18
PyCON UK 2013
Parselmouths dwell in the New Model stronghold of Coventry p50
Qt 5
What next for Nokia's lost orphan child? p54
Desktops
Customise a GUI
» Build a bespoke desktop environment
Websites
Drupal
» Unleash world-beating Web 3.0 sites the easy way
Load balancing
HAProxy
» Beat the Slashdot effect with high availability
TuxRadar.com

Fig. 1 The benefits of subscribing

Save up to 40% per year
Get thousands of DRM-free tutorials, features and reviews



Hello *Linux Format* reader!

Is this the first time you've read LXF?

YES

Read, enjoy, come back next issue.

NO

Do you buy LXF more than a couple of times a year?

YES

Are you a subscriber?

YES

Well done! You've saved yourself the hassle of getting down to the shops – town centre parking is a nightmare these days.

NO

Don't bother subscribing; it's not worth it. Instead, write to us and let us know what you want to see more of and what you want to see less of. We want your cash!

NO

Are financial incentives important to you?

YES

Do you live in Foreign and visit the UK more often than once a month?

YES

Buy it on the newsstand in the UK and save our costly export fees.

NO

You should subscribe. At just £54.99 a year for UK readers, it's far cheaper than buying each issue individually.

YES

NO

Congratulations on transcending materialism. Perhaps you'd like more than 2,700 DRM-free PDFs covering everything from *Asterisk* to *Zshell*, available exclusively to LXF subscribers.*

NO

Do you own a TARDIS?

NO

You should **subscribe**. You'll get your copy up to a week before it hits the shops, and be able to download a digital version.

YES

You are the Doctor, and I claim my prize. Also, please take me back to 1968 so I can see the Elvis comeback special!

*That's all of our back issues, starting with issue 67 and going right up to the magazine you're holding in your hands.

Visit www.myfavouritemagazines.co.uk/LINP2B

(Please use the full web address to take advantage of this special offer)

or call our hotline on 0844 848 2852 and quote LINP2B
For international subscription offers go to:
www.myfavouritemagazines.co.uk/LINP2B

In the USA or Canada?

» Subscribe for $139 a year, or $35 every three months
Go to www.imsnews.com/lxf-a049
or call TOLL-FREE on 1-800-428-3003 and quote 'a049'

All savings are based on current newsstand prices. Payment is in GBP unless otherwise stated. You will receive 13 issues of *Linux Format* a year. If you are dissatisfied in any way you can write to us or call us to cancel your subscription at any time and we will refund you for all unmailed issues. International subscribers please allow up to eight weeks to receive your first issue. Access to the subscriptions-only archive area is strictly for personal use and not available to digital subscribers. Offer ends 15 December, 2013.

Add your voice to the Sound of the crowd

You don't need a profit-minded VC to invest in your ideas. Instead follow **Mayank Sharma**'s advice and raise money from people passionate about your project.

What's the common link between a customisable smartwatch, an Android-based gaming console and an open source blogging platform and an open world survival videogame? Besides all them being hugely popular in their respective genres, these compelling hardware and software projects all saw the light of day only thanks to small monetary contributions supplied by complete strangers.

The idea of crowdfunding or reaching out to individual people to raise money isn't new. What is new is the way the money is being raised. Thanks to the internet and social media, it's easier to find and connect with people who share your passions. This leads to more effective and efficient ways of reaching the right people who are willing to open their wallets to fund your projects.

Crowdfunding is nothing short of an economic revolution. According to research firm Massolution, in 2012 individuals on crowdfunding platforms raised $2.7 billion and successfully funded more than 1 million campaigns. By the end of 2013, the firm expects the average joe investor to take this figure to a whopping $5.1 billion.

But while crowdfunding allows you to swim in money, you shouldn't just dive right in. Don't let the super successful campaigns fool you. Harnessing the power of the crowd is an involved process. There's more to raising money than just putting up your story, offering freebies and waiting for the money to roll into your bank account.

Before you launch a crowdfunding campaign, there are a lot of things you should know. A successful crowdfunding project requires all the attributes of a social and a business campaign. You have to supplement your big idea with research and marketing strategies.

In this feature we'll give you the ins and outs of crowdfunding and highlight its advantages over traditional avenues of raising funds.

We'll also share advice from people who have been there and done it to help you plan your project; all the way through launch to successful funding and delivery, while avoiding the potential pitfalls. You'll also learn about the various crowdfunding platforms out there, so that you can make an informed decision when you are selecting the one best suited for your campaign.

Crowd sourcing 101

A crash course in cashing in.

While contributing to someone's crowdsourcing campaign doesn't require much more than a mouse click, setting up your own is a lot more work. Keep these statistics in mind if you're thinking of launching a campaign without doing your homework: only under 50% of all projects on Kickstarter have been successful and 10% of projects that are launched have never received a single pledge!

But once you get the hang of crowdfunding, you can turn those stats on their head. As you'll see it's the best methodology for raising capital and finding a marketplace, whether you are trying to mass produce an innovative hardware device or hack together a piece of software.

What is crowdfunding?

Crowdfunding is a subset of the much broader concept of crowdsourcing, which has come to mean a lot of things, but is the process of connecting with a large group of people via the internet to tap their knowledge, expertise, time or other resources.

Crowdfunding is specifically about raising the cold cash; it's the collective cooperation of people who network and pool their money together to form the financial backing for a project. The crowdfund is an individual or a group that comes forward with a plan for a project they wish to fund, and then look to a group of people coming together to actually provide the money to support the development and actual delivery of the project. Essentially, crowdfunding is using the internet's access to the masses to raise capital by connecting people who have the talent and ideas with people that would be interested in those ideas and have the funds to invest.

There are some key characteristics unique to crowdfunding projects that help distinguish it from other avenues of raising funds and collaborating on projects. For starters, crowdfunding projects involve microfinance. By inviting small contributions, crowdfunding lowers the barrier for people to participate. People who perhaps never thought of themselves as investors before can do so now.

Crowdfunding projects must also offer some kind of reward even if it is something intangible like the mere association with the project or a feel good factor. Rewards make the project attractive and having a diverse range of rewards, often based on the amount that an individual pledges, makes it attractive to a larger group of people.

Also, all projects must have well-defined targets, which is usually a sum that you want to raise within a stipulated time. Another key aspect of crowdfunding projects is promotion and marketing.

The traditional routes for raising money have been through venture capitalists and angel investors that offer private equity or through plain-old monetary loans from your local bank. An eager entrepreneur approaches these

»

Popular crowdfunding platforms

So you've done all the legwork and are ready to approach people for funds. Where do you do it? Do you put up a website and set up the complex infrastructure to process credit cards or Paypal? You could surely do that (*see the list of DIY Crowdfunding Platforms, starting on page 39*), but it's much easier to set up your campaign on an existing platform.

Using a platform offers several advantages. For one, they have the necessary functionality to carry through your campaign. They'll host all your media and give you the right tools to engage with your audience. You'll also have access to various social media tools to help promote your project. Also a crowdfunding platform puts your project in front of a crowd that's looking to put up money.

Bear in mind that they will all also charge you a different fee for using their service, and some might also ask you to pay the processing fee for the mechanism you use to withdraw the money you've raised. Here's our pick of the current crop.

Kickstarter

Web: www.kickstarter.com
Funding model: All or Nothing
Fees: 5% if you meet your goal, plus payment processing fee.
Payments: Amazon Payments

About: The 800-pound gorilla in crowdfunding. Only hosts campaigns in particular categories and has an approval process. Also has lots of information to create and run a successful campaign.

IndieGogo

The world's funding platform. Fund what matters to you.

Web: www.indiegogo.com
Funding model: Keep it All or All or Nothing
Fees: 4% if goal met, 9% if you don't, plus processing fee.
Payments: Paypal, ACH, Wire Transfer, or FirstGiving

About: *Hosts campaigns worldwide across all categories. It's more expensive if you don't reach your goal. Also has lots of documentation.*

Fundable

Online Funding Platform for Small Businesses

Promote your business with a beautiful company profile

Allow investors to support your company for as little as $1

Streamline fundraising with online transactions and promotion

Web: www.fundable.com
Funding model: All or Nothing
Fees: $99 per month during active campaigns plus payment processing fees.
Payments: WePay

About: *A platform especially designed for funding startups that can offer rewards or equity in exchange for funding. According to the website, startups that offer equity typically raise up to $10 million.*

entities with a business plan and has to hope that the person scrutinising every detail of their plan can see that there will be a good financial return for any initial investment.

This is in stark contrast to crowdfunding where, using online social media, the entrepreneur can access hundreds of potential investors through crowdfunding websites, who aren't necessarily interested in making a buck off the back of someone else's talent. This is what makes it one of the most popular non-traditional methods of acquiring funds.

The barriers are down

Unlike the usual funding methods, where the funds come from a single entity or a tightly knit group, a crowdfunding campaign leverages social media to pitch the project to the mass-market for achieving its financial goal. Here modest cash contributions from a lot of people add up to a substantial amount of money.

The most exciting thing about crowdfunding is seeing it in action: "The thing about crowdfunding is that it totally removes all the barriers between the person with an idea and their customer," believes James Carey, designer at Big Robot, developers of *Sir, You Are Being Hunted* which was funded using a Kickstarter (**www.kickstarter.com**) campaign.

"You don't have to convince anyone other than the end user that this thing is worth having. No banks, no focus groups, no grants councils. Because of that, ideas that would never have been seen as viable by 'experts' in a given industry get the chance to exist. In my opinion that's the best thing the internet has done in years," says Carey.

Depending on the type of campaign you are running, people that contribute to a crowdfunding project don't expect a monetary return, shares or even their money back. Instead of treating it as a traditional form of investment, they invest to support a cause or venture they believe in, with the hopes that it will succeed. Carey is of the opinion that one should look at crowdfunding as purchasing, not investment: "People are speculatively shopping. Buying in advance. Pre-ordering."

Another factor that sets crowdfunding apart from the traditional sources for securing funds is the passion of everyone involved. You are pooling funds from people who are passionate about a project or an idea and want to help bring those ideas to life.

In a webinar, one of the oldest crowdfunding platforms, Indiegogo (**www.indiegogo.com**) advises people to think of crowdfunding as shared enthusiasm and not panhandling. The whole idea of a successful crowdfunding campaign should be to share the passion that you have for a specific project in order to get the audience excited about it.

You are allowing them to look behind-the-scenes and understand why you are so passionate about the project in order to get them onboard and contribute. No matter how much funds they may or may not have. If they don't share your enthusiasm, they won't contribute.

The side perks

Although raising capital is the most obvious reason for crowdfunding, there are several other ancillary benefits to following this route with your project. Running a successful crowdfunding campaign can give you a fantastic opportunity to gain visibility and build interest in your project. You not only get to gauge the demand for your project, you do so with minimal financial risk in a very short span of time as compared with taking onboard venture capital by offering stakes in your startup. The Ubuntu Edge campaign on Indiegogo is a classic example of this. Canonical didn't reach its hefty target of $32m (and by reaching over $12.8m the campaign broke crowdfunding records) but that didn't stop the mainstream press covering the fact that an Ubuntu OS was coming to smartphones in 2014.

In essence, running a crowdfunding campaign will let you know if you have a good idea and if there's a demand for it in the marketplace. It isn't surprising then that many successful campaign managers have gone back to the venture capitalists that had initially turned them down to show that there's a demand for their project and that people are willing to pay the price they had set.

Crowdfunding models

What you've read up to now is a very broad view of crowdfunding. In practice crowdfunding encompasses several different fundraising models.

To begin with you have the donation model that has been around for years. People donate money to a project or cause they believe has moral and ethical value and is good for the community. They don't expect any return for their financial contribution besides the satisfaction of having made a difference.

Then there's the rewards-based model. This is the model which comes to mind when most people think about crowdfunding. In this model people who are making a financial contribution can pre-buy a product or are rewarded in some way. So the crowd makes a monetary pledge to the project and the project offers them something in return.

This model is further subdivided into two models.

Crowdfunding rules and regs in the UK

To grasp the legal issues of crowdfunding in the UK we had to turn to our in-house legal expert, Shashank Sharma.

The UK's first regulated crowdfunding platform, Abundance Generation was approved in 2011 and launched to the public in 2012. Back then crowdfunding platforms in the UK were being regulated by the quasi-judicial body known as the Financial Services Authority (FSA). Since April 2013 that task is now handled by the newly constituted Financial Conduct Authority (FCA). According to the FCA, there's no real fraud protection for investors interested in crowdfunding. While investors can turn to the Financial Ombudsman Service (FOS) or Financial Services Compensation Scheme (FSCS) in case of fraud, these services are available only when the platform is authorised by the FCA or its predecessor, the FSA.

There are, however, several platforms that are operating without such approval, and the FCA estimates that the number of unauthorised platforms actually far outnumbers platforms that it's approved.

In the US, equity crowdfunding wasn't made legal until the passing of the JOBS (Jumpstart Our Business Startups) act in 2012. More recently, on September 23, 2013, Title II of the JOBS act went into effect. While the JOBS act enables startups to seek funding through crowdsourcing, Title II gives them the freedom to use sites like Twitter and Facebook to spread the word about their campaign.

The first is the 'all or nothing' model where a project has a specific financial target that it sets and intends to reach. If the campaign doesn't reach that target the project initiators do not receive the funding, and the lenders don't receive their rewards.

The other model is the 'keep it all' model where whatever money is raised in the crowdfunding campaign is retained by the project initiators irrespective of whether they reach the set target or not. Many consider the 'all or nothing' model to be a more compelling choice.

There's also a lesser-known "bounty" model where the accumulative funds collected for a particular task are given to anyone who completes the project.

Another less high profile, yet revolutionary, model is the 'equity-based' model which allows people to invest small amounts towards funding startups. In exchange they can expect to receive dividends based on the profits of the business. This is more popular in the UK and Europe because of there are less legislative hurdles, (*see Crowdfunding rules and regs in the UK*) but the model may soon be legalised in the US as well.

Running a successful campaign

There are quite a few differences that separate successful campaigns from others that fail to reach their target. Before you launch, prepare every aspect of the campaign in as much detail as you can. You can't fly by the seat of your pants. You need to make sure you have a plan for every week of the campaign.

James Carey says that your preparation also shows that you have already invested some of your own time and money on a project: "We didn't want to go to Kickstarter until we already had a solid prototype, based on around six months of work and the considerable investment of cash that that implies. If you aren't prepared to risk it for your project, why should your backers?"

One of the first things you should work on is your campaign pitch. Browse successful campaigns and check out why they are compelling. Most pitches are very personal and clearly specify who is behind the project and what problems they hope to solve with their project. They also tell backers how they can get involved.

"Pitches succeed if people judge they are going to succeed," says Carey. "Something that looks like it will get its money generally does. It's a perception game."

A good engaging pitch introduces the campaign both in writing and in the form of a video. According to stats shared by Indiegogo on its blog, campaigns with pitch videos raise over double the amount of those without one. Also keep in mind that the average campaign video length for successful campaigns is roughly 3 minutes.

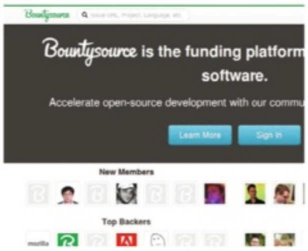
It's also imperative that you set an attainable goal. Don't just set an arbitrary figure. Do some serious number crunching and set a conservative goal that'll help you move your project forward and also fulfil the promised rewards.

On Indiegogo, 87 per cent of campaigns that reach their goal exceed it by an average of 32 per cent. While advising projects to have reasonable targets, Carey notes there seems to be a pattern: "An interesting rule of thumb, actually, is that successful Kickstarter projects get 150 per cent of their target, generally. This suggests that if your target is £60k, you should ask for £40k, and be more likely to get it all."

»

Popular crowdfunding platforms continued...

BountySource


Bountysource is the funding platform software.

Accelerate open-source development with our commu

Web: www.bountysource.com
Funding model: Bounty or All or Nothing
Fees: 10% non-refundable fee for placing bounties
Payments: Paypal, cheque, Google Wallet, , wire transfer

About: *Designed for crowdfunding open source software. Hosts individual bounties to resolve open issues and feature requests, and also used to raise money for big updates or new projects.*

Freedom Sponsors

Crowdfunding Free Software, one issue at a time

Place money bounties for open source projects **issues.** Learn more...

Sponsor/Propose a new issue

Issue URL

Web: http://freedomsponsors.org
Funding model: Bounty
Fees: 3% + payment processing fees
Payments: Paypal and Bitcoin

About: *Enables many users to chip in to a bounty. Bounties are post-paid only after the sponsors have verified the work. Uses Paypal's parallel payment type to split bounties between devs.*

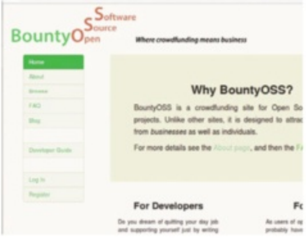
Catincan

Fund New Features
on Your Favorite ❤
Open Source Projects

Web: www.catincan.com
Funding model: All or Nothing
Fees: 10% after funding amount has been reached.
Payments: Paypal, Bitcoin, wire transfer

About: *Only allows developers on existing open-source projects to create campaigns. Features are screened and developers have 60 days to reach the funding goal else the backers aren't charged.*

BountyOSS


Where crowdfunding means business

Why BountyOSS?

BountyOSS is a crowdfunding site for Open So projects. Unlike other sites, it is designed to attrac from businesses as well as individuals.

For more details see the About page, and then the F

For Developers

Do you dream of quitting your day job and supporting yourself just by writing

Web: https://bountyoss.com
Funding model: Bounty
Fees: 5% + payment processing fees
Payments: Credit Card or bank account

About: *Designed to attract funding from businesses and other for-profit entities that use open source software. Only allows developers attached with an open source to create campaigns..*

You should also carefully devise your rewards. Make the rewards unique enough to get people excited about them and want to tell their friends, while not breaking the bank. Again, it's best to learn from some of the successful campaigns of the past, many of which offer between five and eight rewards.

It's also best to have a wide range of rewards. Instead of unveiling them all at the same time you can update them during your campaign. Again it's best to browse campaigns and look for perks that have been claimed a lot. John Nolan, who successfully funded the open source blogging platform Ghost, shares his recipe for success: "A compelling idea, value for money on rewards, and a clear indication that the project creator is able to deliver."

Foster a community

Once you've worked on these crucial elements also spend time working on your communication plan. Successful campaigners are proactive communicators throughout the life of their campaign and keep their investors engaged and excited. While some treat the crowdfunding platforms as a blog, most have 1 to 2 updates a week. They'll discuss updates, milestones and new rewards as they are unveiled.

James Carey believes that updates are key to the success of a campaign and adds "You've got to foster a good relationship with your backers by letting them know what you're up to behind the scenes. Crowdfunding is patronage, but there's a certain sense of ownership that comes with that patronage. You're in this together with your backers, let them know it."

Finally and most importantly, don't forget to hone your social media skills. What you are looking to achieve is a large number of small donations which are aggregated together to produce a significant sum. To do this you need to be able to reach and pitch your project to a large audience, and social media is the perfect tool for extending the reach of your campaign.

Using social media and the idea of six degrees of separation, you can rely on members of your initial core community to reach out to their respective communities and ultimately create a viral distribution campaign.

Also remember that the initial days of the campaign are very crucial. You are solely responsible for generating the initial momentum. This is why it's important to identify your initial target audience; the ones that will jumpstart your campaign and become early contributors and promoters.

Crowdfunding comes with some potential pitfalls. So before you launch your campaign you will want to learn from the mistakes of crowdfunding failures to avoid turning off investors and customers.

According to Tony França, creator of FreedomSponsors, there are two factors for success on a crowdfunding campaign: "Some projects fail basically because they have no market, and many fail because their marketing is not good enough. Your idea must actually be something that people need or want. Otherwise you have no market and your project will fail. Period. And even if you do have a market, you really need to work on the marketing. The world needs to know about your campaign. This is the hard part. I believe the most successful campaigns are the ones who market their product so well that it makes people want to tell others about it, and then they go viral."

Warren Konkel, CEO of Bountysource, points to another reason for unsuccessful campaigns that haven't done any background work and research: "Generally speaking, fundraisers fail to meet their goal for one of two reasons: people don't want it or people don't understand it."

The bigger problem, however, is projects failing to map their financial strategies. If you haven't done the initial research, don't be surprised if the total cost of the project and rewards end up exceeding the amount you raised.

One of the key factors for faltering campaigns, according to Nolan, is pointless rewards that have excessive fulfilment criteria: "You're going to personally sign 10,000 postcards? Really?" Launching a campaign and setting a goal thinking that you can 'figure out the tax stuff later' is another pitfall that he highlights.

Carey agrees that some projects promise too much without properly costing their physical rewards: "I think delivery costs have bankrupted a couple of projects because they didn't put a cap on how many branded mugs they had to deliver."

The Hanfree iPad Accessory is a prime example. The project creator underestimated the real cost of bringing his idea to market and mass produce a product. After successfully meeting the $35,000 goal, he couldn't deliver the product nor the rewards. Eventually one frustrated backer sued the project creator who then declared personal bankruptcy.

Ready to roll?

By now you must be itching to get started on your precious project. But before you do, here's a quick summary of what's involved in launching and running a good crowdfunding campaign. There are a bunch of things you must decide before you can even begin looking for a platform to run your campaign on. First, decide the target you want to achieve while keeping in mind the nature, number and diversity of the rewards. Also, spend time researching what duration to give your campaign by

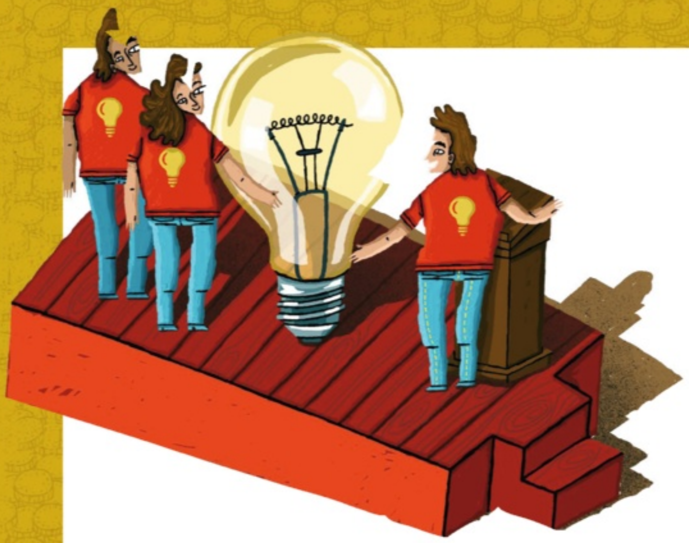
Ideal timeframe for a campaign

There is a correlation between the duration of a campaign and the amount of money it generates – but it isn't what you think. If you think the longer you run your campaign the more money it would generate, you'd be wrong.

According to Indiegogo, the majority of successful campaigns run on average for about 30 to 40 days. This makes the campaign short enough that a tangible end is in sight and long enough to gain momentum and engage a wider audience as well.

Kickstarter's research also points to the same conclusion; projects on Kickstarter can only last up to 60 days, because according to its studies, projects lasting any longer are rarely successful. Instead, it recommends that project deadlines are set at 30 days or less because shorter durations create a helpful sense of urgency around the project.

Besides, you can also prepare for a shorter campaign more effectively. You will also find running a campaign is a lot of work and for many projects it's full time task for the duration of the campaign.



> “You’ve got to foster a good relationship with your backers.”

looking at similar successful projects. Picking a long duration may seem like the simple answer, but remember you will have to keep the momentum going for its entire duration (*See Ideal Timeframe for a Campaign, p39*).

When you’ve worked out these bits, start looking around for a platform. In the last couple of pages of the feature we look at some of the best crowdfunding platforms and tools. Decide on which funding model is ideal for you and your project.

Promotion planning

While you’re at it, also think about the different social media tools you’ll be using to promote your campaign. Using a wide range of social media tools might seem like a good idea, but only do it if you can manage to keep track of all of them. For instance, an unanswered query from a potential investor can do a lot of harm.

Also make a list of your initial target group. Besides contacts in your network, remember to reach out to the mainstream media and blogs that have audiences with a similar interest. It’s also a good idea to inform them a couple of days before you actually launch your campaign, and hit them again once your campaign is live.

Don’t forget to keep exploring different avenues for promoting your campaign even after you’ve launched it. A crowdfunding platform is not eBay. Your campaign won’t sell itself. Also remember that you aren’t only looking for people who’ll back your project monetarily; you are also looking for co-promoters that will introduce your project to their network.

Keep your investors updated during the duration of your campaign. Share your excitement, new ideas, new rewards and even ask for their opinion or advice if there’s scope for it. Whatever you do just make sure you keep them involved. When you are funded make sure you inform your backers about the status of their rewards, and get cracking to bring your project to life.

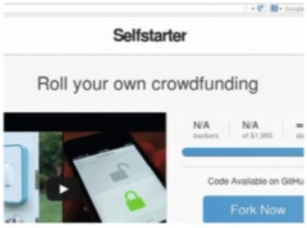
»

DIY crowdfunding platforms

There are plenty of reasons for projects to not use an existing crowdfunding platform and rather roll out their own. For some it’s purely ideological. Can you imagine the FSF running a crowdfunding campaign on a closed-source proprietary platform?

But there are some pragmatic reasons as well. As larger platforms become more crowded with projects, it negatively impacts their visibility and discoverability. As crowdfunding gains acceptance as a legitimate mechanism for raising money, if you can bring your own crowd you can raise money without relying on a well established platform. One of the most successful examples of a DIY platform campaign is that of Lockitron. The project was rejected by Kickstarter so they set up their own crowdfunding infrastructure and raised over $2 million. They then released their platform as open source software for others to use.

SelfStarter



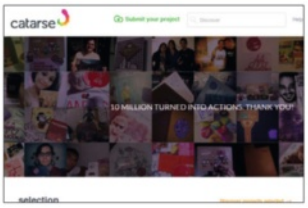
Selfstarter

Roll your own crowdfunding

Web: http://selfstarter.us/
Licence: MIT License
Fees: Free to use
Payments: Amazon Payments, Stripe, WePay

About: *Developed using Ruby on Rails, the platform can be extended with custom authentication, administration and product management code. The developers suggest deploying it on Heroku application platform.*

Catarse



Web: https://github.com/catarse
Licence: MIT License
Fees: Free to use
Payments: MoIP, Paypal

About: *Another software built on Ruby on Rails. You can use it to create your own crowdfunding platform to host several campaigns. Powers the Brazilian platform* **http://catarse.me/en**. *Has minimal documentation but an active mailing list.*

Goteo

Web: https://github.com/Goteo
License: GNU AGPL v3
Fees: Free to use

Payments: Paypal

About: *The software is written in PHP and facilitates communication between users and campaign managers as well as the creation of blogs, and static pages. Besides the English installation guide, the other developer docs are in Spanish.*

Crowdfunding open source software

Thanks to the nature of free and open source software, most of the software development is crowdsourced, but will it become crowdfunded too?

As it turns out, the most popular crowdsourcing models offer a unique challenge to open source software. Although you can find successful campaigns for open source software on popular crowdfunding platforms such as Kickstarters, they aren't a natural fit.

In an interview with **opensource.com**, Warren Konkel, CEO of Bountysource, a crowdfunding platform designed specifically for financing open source software, says that free and open source software needs a different treatment than the usual wares on offer at other popular platforms.

"Those other platforms work well as a pre-sales model for physical consumer goods and technologies, but we believe open source software needs a better funding model that's more aligned with how software is built."

Talking to **LXF**, Konkel explains this further by saying that one of the reasons that sets software development apart from hardware is that software development is notoriously hard to estimate. "The Bountysource platform solves this problem by associating fundraisers directly with existing bugs and feature requests."

Another challenge for crowdfunding open source software that decides to use platforms with a reward-based model is selecting the right kind of rewards to tempt pledgers in. A typical proprietary software campaign can offer various versions of the software as a reward. In contrast, simply because of its open nature, open source software cannot offer product versions with special functions available only to funders.

What they can offer are mainly ancillary services like personal support. Some give backers credit on the project's website, some provide exclusive content such as a regular funders-only newsletters, and even physical products, such as exclusive campaign T-shirts.

As always, looking at what other open source software campaigns are offering will give you an idea of what to offer. The Ghost blogging platform offered free accounts on its hosted service, promised backers early access to the platform and offered them a chance to reserve their

Know the lingo

Here's a low-down of the popular crowdfunding jargon. If you are looking to create a campaign you'll run into these words regularly.

Campaign: These are the projects that you are looking to fund.

Backers: Also referred to as "Contributors" and "Funders" these are the individuals who support your campaign by giving you money.

Goal: The amount of money the campaign is looking to raise in the time specified.

Stretch goals: If a campaign reaches its goal prior to the deadline, it can introduce a new deliverable to the campaign that'll be worked on if the new higher set goal is reached.

Pledge: In an "all or nothing" model, until a project meets its goal, the monetary contribution is called a pledge.

Rewards: Also referred to as "Gifts" or "Perks" these are the incentives offered to backers in exchange for their financial contributions.

username on the community website, which was displayed with an emblem to acknowledge their support.

Many also see crowdfunding in open source software development as part of a long-term process rather than a one-off investment. Konkel says that at Bountysource its looking to create long-term relationships between developers and backers: "When somebody backs a fundraiser, chances are they'll back a subsequent fundraiser or create a bounty."

Philip Horger who is an active fundraiser at FreedomSponsors, another crowdfunding platform for open source software, shares an interesting experience of this developer-donor engagement.

Horger put up a sponsorship to improve the user interface of *LibreOffice*. A member of the actual development team for *LibreOffice* showed up and advised that sponsoring such a wide-scope issue was, while an encouraging sign for the developers, not a realistically useful way of making UI improvements. The developer, instead, advised Horger to sponsor more specific improvements, such as the colour picker, etc. "So I broke up my offer among more specific sponsorships. The total sponsorship was the same in the end, but more usefully distributed."

Better than donation

Monetary contributions from users for supporting open source development isn't something new. But crowdfunding is much more gratifying than a simple donation.

Horger says that sponsoring on FreedomSponsors is inherently more fine-grained and personal than donations since an offer is tied to a specific and achievable outcome: "You feel that you have stake and a responsibility in that sponsorship's success. It is more gratifying and tangible to be able to point to some distinct feature of some popular program, and say to the person next to you, 'I helped make that happen.'"

Another advantage of platforms, such as FreedomSponsors, according to Horger, that doesn't really make sense for traditional donation is the ability to sponsor one's own projects. It might seem strange at first but as Horger puts it, "there's a method to my madness."

"Sponsorships are my way of motivating/rewarding external development, and that includes starting with small stuff. That way, I can foster an active and healthy community of contributors, which not only will take programming and maintenance load off of me in the long run, but also means that my projects can easily live on if I get hit by a bus."

The future of crowdfunding

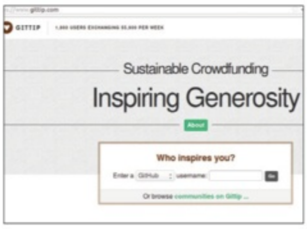
Like his peers, Tony França of FreedomSponsors is very gung-ho about the future of crowdfunding and how it will help developers generate funds: "We are witnessing the birth of new businesses powered entirely by the crowd. I believe Crowdfunding will become more and more prevalent as a way to gain access to capital."

Konkel also acknowledges the power of the fundraiser model especially for projects that require upfront capital to achieve economies of scale and where rewards are effectively pre-sales. But he believes that it won't work well for everyone. "We think the future of crowdfunding in open source is a bounty-based model: placing individual bounties on existing bugs and feature requests. Issue trackers can quickly become overwhelming and bounties allow backers to focus development efforts on the issues that matter to them." LXF

DIY crowdfunding platforms continued...

GitTip

Web: www.gittip.com
License: CC license
Fees: Free to use
Payments: Bitcoin, credit cards



About: *Gittip is a crowdfunding platform to support people by giving donations weekly. You can use the software to setup your own weekly gift exchange for the various individuals involved in your project or community.*

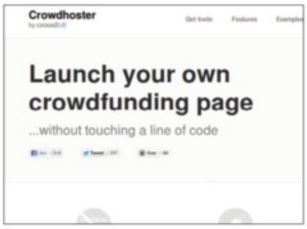
Spot.Us

Web: https://github.com/spot-us
License: GNU GPL
Fees: Free to use
Payments: Paypal

About: *This Ruby on Rails project lets you setup a crowdfunding website to enable individuals or a group to solicit funds for commissioning freelance journalists. The project has minimal document but you can see it in action on* **www.spot.us**.

CrowdHoster

Web: www.crowdhoster.com
License: MIT License
Fees: Free to use
Payments: Credit cards, Bitcoin


Crowdhoster
Launch your own crowdfunding page
...without touching a line of code

About: *Proudly calls itself the "Wordpress for crowdfunding", the hosted project is currently invite-only. It's free and pretty straightforward to setup and administer since there's nothing to install. Also includes tools to manage contributors.*

Ignition Deck

Web: http://selfstarter.us/
License: Proprietary
Fees: Starts at $79 (£49)
Payments: Paypal, Stripe, WePay

ignitiondeck
A Powerful Wordpress Crowdfur
Crowdfund, Pre-Sell or Raise Money For Yo
STAR CITIZEN SUCCESS STORY $8 MILLION RAISED WITH IGNITIONDECK
SIRIUS

About: *This tool is in fact a Wordpress plugin. It's available in several version starting $79 (£49). You can install the plugin on multiple WordPress installations and buy additional themes and features such as payment gateways and an analytics panel.*

Renaissance man

Graham Morrison meets Robert ‘r0ml’ Lefkowitz, a man who crams 2,000 years of open source philosophy into 3,000 words.

Robert Lefkowitz, who is nearly always referred to as ‘r0ml’, is an old skool hacker, programmer and thinker who switched from studying nuclear physics to computer programming in the 1970s. But the more words we type here, the fewer we can print from the man himself, so let’s get to it.

Linux Format: **Do you think computers and programming languages should be easier to understand?**

Robert Lefkowitz: So Charlemagne was concerned, in 789, he has this capitulary called The Adminitio Generalis, in which he reasons that monks learn how to read, so that they can read the word of God, so that they can follow God’s law more effectively. If that’s the case, why wouldn’t you want every boy to learn to read? In fact, in this capitulary he says we will establish schools to teach every boy to read, freeman and serf. That, as near as I know, is the first aspiration to universal literacy.

The other one is what the Scandinavians did a couple of hundred years later, if the emperor in those times puts out a capitulary, how is everyone in the empire going to follow it effectively – at least one person in every village has to know how to read it so that everybody will have roughly the same interpretation. So it’s a way of creating cohesion for the empire. But the difficulty was that the technology was so bad that it was very difficult to read and required a lot of effort and training.

My personal favourite is one of the things Alcuin (of York) shortly after invents is spaces between words.

LXF: **Why weren’t we taught this in school!**

RL: It’s one of those things that everyone takes for granted, “Obviously you’re going to have spaces between words.” But especially in Latin, there are a lot of phrases and a lot of mistranslations because if you look at the sequence of letters, you could break it in any one of two places and get two different things – and which one is correct? That’s why they illuminated manuscripts because reading was so complicated. You needed pictorial hints so you knew what it was about.

LXF: **They added images before spaces?**

RL: And every scriptorium, since these things were hand-done, they each had their own hand, so there was no standard font. So, York comes down and he comes up with this standard font that’s called the Carolingian minuscule to which all modern fonts trace their ancestry. He invents things like standard height ascenders, standard height descenders lengths, and then you add spaces between words and all of a sudden you have this ability to see the whole words as shapes.

There’s a whole number of innovations around making it possible for people to read that Alcuin kicked off. The next fifty years or so is called the Carolingian renaissance. This mini renaissance happens before the big famous renaissance later. I argue we’re at that cusp where we’re starting to say. ‘Oh, we should teach everybody how to code,’ but it’s so complicated and difficult, and there’s a thing like spaces between words which in retrospect will be so obvious; that makes it so much easier that we haven’t thought of yet – we need an Alcuin of York.

LXF: **We’re waiting for the spaces between words… it’s not Python then?**

RL: (laughs) Well, I gave this talk out of PyCon, in fact, where I said a big revolution is to make spaces significant. So the Python audience can certainly relate to the concept that maybe that’s the thing. But it was more than just the spaces between words. So maybe Python is that thing, but it’s one of those things that’s difficult to know except in perhaps a 100 years later when we look back…

LXF: **It’s the concepts, conditional statements, the effort of constructing a solution to a problem that stays with you. This is what non-programmers are missing. It’s the demystification of the process that means a lot of people go through the world thinking computers are magic. How do we tackle that?**

RL: I certainly think it’s a more conceptual kind of education. I had set out about a year ago to write a book on this topic. And I had this thesis around what we’ve just been talking about – around Charlemagne, and this kind of cusp when you make it easy enough, universal literacy, and how universal literacy for computers is sort of analogous to universal phonetic literacy and what should we do to move that forward. But doing my research, my thinking evolved. I came across some other interesting stuff that I read which made my thinking evolve to the point where I don’t believe my thesis holds water!

LXF: **What about the book!**

RL: There’s still going to be a book! But I’ll delay the schedule now because what I have to do is think about it some more and come up with a new thesis that I think holds water better, and that would be more effective. Fundamentally, the idea that literacy, from Charlemagne, and sort of subsequent revolutions of it throughout history, there were two main drivers for literacy and education, and literacy and education was all around reading. The reason it was all around reading was for these two societal myths; the first one was the Christian myth: in order to know the word of God you had to read the word of God. To read was to become better. You read in order to become good, and that was kind of the societal driving force. If it’s going to make people better then obviously everyone should listen. And then the second myth, for us in colonial America, but also throughout the world, in order to have an effective democratic society, or any society, you need an informed citizenry. We felt we needed this because they needed to vote. In the non-democratic societies you needed to have an informed citizenry because they needed to know which laws to obey without getting in trouble. But in order to have an informed citizenry they needed to know how to read, and therefore, you have to teach them all how to read. Ta da! Because it will make them better citizens. »

ON TEACHING PROGRAMMING

> **“There’s a thing which in retrospect will be obvious – we need an Alcuin of York.”**

Reading, is in Deborah Brandt's [professor emerita of English at the University of Wisconsin-Madison] words, 'for good' not 'a good'. Writing, however, was 'a good'. You wrote, and it was a product, and then you could sell it. You didn't improve yourself by writing ... That's the myth. And if you look at computing science in the 1980s, Donald Knuth's *Literate Programming*, they call talk about 'Yes, you have to write programs that can be readable so that other people can read them'. It's all informed by the underlying cultural sensibility that says 'Oh yes, reading is a good thing'. What I'm saying is something different.

The second piece of the puzzle is the thesis that I'm developing now. In the beginning nobody reads and writes because we haven't invented that stuff, or it exists but it's only for bookkeeping. It doesn't have any societal impact because its all scribes in the royal treasury. Then, and we'll take this from Plato, Socrates believed writing was a bad idea. Plato agrees with Socrates but he writes down *The Republic* to explain why it's a bad idea.

LXF: We're covering a lot of ground here... from the 1980s to 400BC.

RL: (much laughter) Then from Aristotle, Plato, on, you have few readers but it's a societal thing, and few writers. A smaller number of people read, a smaller number of people write. This holds true roughly until Gutenberg – the printing press. They call that manuscript culture. In manuscript culture you don't have the notions of copyright, originality, plagiarism, because there's so few readers that it doesn't matter. The idea is that you couldn't possibly come up with a text all on your own anyway, because that's such a complicated thing, and you're pretty much pulling together from other texts and recombining stuff that's already there and you don't have to do attributions – none of that stuff, none of it exists.

LXF: These texts would be amended as they were copied?

RL: Either on purpose, or accidentally, yes, both. Sometimes there would be transcription errors – and there's some fascinating stuff when you read up about it. Sometimes it was anthologising, you'd be saying 'this is a good thing', and this one and you'd put them together. Sometimes you'd bother to mention the original source was and sometimes you wouldn't – what difference did it make?

LXF: When no one could Google the source...

RL: Right. So you have this notion of intertextuality. That means texts come from other texts. You can't really create a text from scratch. So you have that manuscript ethos and since there are really no readers, there's no commercial interest in writing or reading. Then you have the printing press and now you have a mass readership. This is what turns writing in 'a good' and reading into 'for good'. You still have a small number of writers because you have machinery in order to do the distribution, access and expense. But there is also now commercial gain. If you write something you can make money, and if somebody else 'steals it', that's where you have the notions of copyright, plagiarism, authorship – even the idea of authorship. Like who's the author of this scroll, you wouldn't even write your name down, because why would anybody else care? You didn't care, they didn't care, nobody cares.

LXF: The writer isn't even thinking about it.

RL: ...So now what I'm suggesting is that there's a fourth age. The fourth age is Dave Winer, right, blogging happens, texting happens, Twitter happens. We're all writing more. Most people today write more than Dickens did in his lifetime. Not as good as Dickens, but more than Dickens! It's the volume. People say you can't make the comparison, because people write mostly lolcats and crap stuff and Dickens was awesome. But if the volume increases one thousand fold. If the average quality drops 50%. You've still got way more high quality than you used to. The level rises.

We had this big ratio of readers to writers and now it's reverting back to its [dramatic pause for effect] medieval ratio. It was few and few but now it's many and many. When that ratio goes back, what should happen? And this is where I haven't thought this through, but in some ways it's going to revert to the medieval sensibility, and maybe that's why you see the weakening of the ideas of copyright, and it's not just happening with software, it's happening with music and journalism. You have this sort of disruption, not just in the business, but in whole societal/ethical framework around what's right and what's wrong. And that's being challenged. But the second thing that happened, and this is the one that's specific to code, which I think changes it so that's it's not just a reprise of the medieval era with the bar moved up, and that is – and I got that from Project Euler (**http://projecteuler.net**).

Project Euler was a spin-off of a math learning site, it got very popular, and it got about 450 problems which are all math problems, because it started out to be math education, but they were the sort where you could only sort them with a computer. So the first problem is very simple – what's the sum of all of the numbers less than a million that are easily divisible by 3 and 5? You could do it with paper and pencil, but you write a computer program to do this, and the idea is that they get progressively harder. The thing that triggered this for me is that a FAQ on the website says: "Oh, I've figured out a really great solution in my favourite programming language to this problem, can I post it on the internet?" The response is, "You've answered your own question." You know that feeling you get that you're so proud of yourself because of a great solution you've figured out – don't steal that from everybody else. Don't post your code.

Here what we have – and granted it's a very specific and narrow case, but I'm thinking from Aristotle on so I've got a thousand years forward to play with here, this is a case where somebody is saying, "To publish your source code would be wrong. Ethically wrong." Combining this with what we were talking about literacy, it's the writing in this case. When you go to Project Euler, you don't learn by reading, because you could – that's the old style of education, I'll just read up on these

algorithms and that will make me a better person: no, no, no, no. The way you're going to figure these out is you're going to sit down and you're going to try to write solutions, because the thing that the computer does, the difference in terms of literacy, is that if you write something, how do you know if it's crappy or not? You have to get somebody to read to get any kind of feedback.

When you write a programme, the computer will give you feedback. That might not be in a legible style for other humans but we're getting that medieval holdover stuff going on here. It will tell you the algorithm is more efficient or less efficient than that other thing you tried, and it will tell you whether you're getting the right answer or the wrong answer. So there's some set of feedback that you get by writing it and by writing it again only differently. Just you, just by writing, you become good at that. In this case, writing is 'a good'. Then you see the same thing popping up for Twitter, and Facebook, the cultural push is to say 'You should post, the writing is the good.' Those drivers that said everybody had to learn how to read, and then we start to teach programming, and then when you think about it, how do you teach programming? By teaching people to write a program and they don't teach them to read other people's programs. Is that because of the ethical sensibilities around that, or the practical sensibilities around that? If there were a canon of code, would it be useful for people to read it, and I think not. The cognitive dissonance for me was if I follow this thing through logically, what programming education would look like, open source would be bad in that world.

ON PROGRAMMING EDUCATION
"I think it's more writing driven than reading driven, so it differs from literacy."

LXF: But this is only at a formative stage!
RL: But it was sufficient cognitive dissonance for me to say that I have to think about this some more. The question that kicked this off, and I'm sorry for such a long answer, is this notion of what does programming education look like going forward? I think it's more writing driven than reading driven, so it differs from other literacy education.

LXF: Writing the code should happen at the very beginning?
RL: That's the long-term thinking. Short term, my thinking had been, and I haven't re-evaluated this, had been we shouldn't start with writing because, following a thought experiment – take somebody that you know that doesn't know anything about computers, adults or kids, and give them a program that has already been written in the favourite programming language of your choice, Python, Java, Smalltalk.

LXF: Perl?
RL: Or even Perl, it doesn't matter. Give them a program that is already written. Can they actually run it? Or pass it parameters if it needs parameters? And be able to tell if they need it or not and then be able to figure out how to get the output? Could they actually take an already written program and use it, read it? And I think the answer to that is mostly 'no'. This may be the spaces between words part, which is the 'why is it so hard to just figure out what to do with a program once it's written?' And maybe we should start there. If you're teaching people to write code, even after it's been written they don't know what to do with it, isn't that the cart before the horse? Maybe the first steps ought to be, and this is where open source is a good, 'Go to this website, search for the thing you want to do because somebody's done something that mostly does that, and install it and run it'. You want to do a program that has a blog, you could just go to blogger.com and get an account, but if you want to think like a programmer, you have to go get the code that does that, install it and run it. The programmer would actually write the code, install it and run it, but let's just skip the writing part and see if we can do the rest of it, and which is the harder part? And what does a citizen need to know who's not going to be a professional programmer?

LXF: It should be about equipping ordinary people with the tools to demystify what tools are doing. Does it need this great arc, which is a great thing if you're into computers and you want to learn.
RL: That's exactly right. I agree with you 100%. The demystification so that people understand roughly how an automobile works, so they know it's not going to fly. This idea of just being able to take already written code and see how quick it runs; how it gets used. Something in that space, and writing code is going to be part of that. I don't know if it's necessarily the first part of that, and we always seem to start there.

LXF: That is all we've got, and it's how the people who are teaching were taught.
RL: A large part of what's driving, and this is a holdover I think from the previous age ... when you see people teaching people programming, it's always couched in the terms of you needing to teach people to do this in order to be economically competitive. Nobody says you need to read in order to be economically competitive. You learn to read to be a better functioning citizen. If you approach it from the, "I'm not going to grow up to be a programmer. I want to be a journalist. I want to be a ballet dancer." I wanted to start the book, something along the lines of, because my wife and kids are very much into the arts. I met my wife when she was teaching the circus arts, tightrope walking and so forth. And so, for her job does she need to know how to read? The answer is no. Why should we teach these people how to read? Because you have to know how to read to be able to function in society, right? But it's got nothing to do with their jobs. We don't teach people to read because it will make them more competitive in their jobs, if it's about demystifying computers so that the common man can live in a world that's heavily influenced by automata, they need to understand how it works but they're not going to use it for their jobs. They're going to use it for their daily lives, and how do they do that? I don't know. LXF

Jono Bacon opens up the dusty history books and highlights a pivotal component in the open source machine: the Qt project.

When the Hollywood blockbuster is made of the Linux story, there will be a number of characters playing key roles: the kernel, various desktop environments and certain bootloaders, but there's one character that has stood the test of time, yet endured more drama than it, rightly deserves, and that's the *Qt* project.

Qt is a graphical toolkit and application development platform that has had a long and interesting history, and despite trials and tribulations is now one of the most critical pieces in many Open Source projects.

Today we are going to take a walk down Memory Lane to look at the surprisingly long history of *Qt* and explore where it is today and what the future holds.

Qt, pronounced 'cute' as opposed to 'que-tee', started out life in 1991 when Haavard Nord and Eirik Chambe-Eng began working on the toolkit. As Nord and Chambe-Eng worked on their new project, they formed Quasar Technologies as a home for it, before forming Troll Tech three years later (which would ultimately become Trolltech).

Trip down Memory Lane

Back in these formative days of *Qt*, the duo worked to create a collection of graphical toolkit components. These components comprised of buttons, scrollbars, lists, combo boxes, text-entry boxes, icon views and the other familiar graphical building blocks required for building applications. *Qt* ran on the popular and ubiquitous *X Window System* (at the time *XFree86*) and was written in the powerful, portable and efficient C++ programming language.

C++ was a benefit for many developers. Linux already had a powerful C++ compiler, but C++ also provided Object Orientated Programming (OOP) capabilities that mapped neatly to writing graphical apps comprised of lots of individual pieces that need to work together. As such, *Qt* provided a collection of re-usable C++ objects, with each widget providing a rich API of features and behaviours (known as signals), even going so far as to include convenient classes to wrap around the complexities in C++ (such as the QString string handling object that provides an easier way of working with this data structure in an application).

As Nord and Chambe-Eng provided public releases of their new toolkit, they caught the eye of a number of programmers. Around that time, in the mid to late nineties, the level of functionality that was present in *Qt* was not generally available in most other toolkits, such as *Motif*, the free equivalent *Lesstif*, *Athena*, and the new kid on the block: *GTK*. One aspect of *Qt* that was also interesting to many observers was that the look and feel of *Qt* could be customised by simple theming and as such *Qt* apps just generally looked much nicer than the bulkier *Motif/ Lesstif* alternatives. Anyone who remembers the first version of *Netscape* for Linux will know exactly what I am talking about.

One such developer who picked up on the *Qt* excitement was Matthias Ettrich who was then a student at the Eberhard Karls University of Tübingen in Germany. Ettrich was a passionate Unix and Linux fan but had become disillusioned with the state of desktops for Unix and Linux. He was frustrated that applications didn't look or feel alike and he wanted a consistent desktop and application experience. Ettrich also felt that the desktops at the time were too complicated to use and he craved a simple and easy to use desktop that novice users could understand without having to grow a neck-beard. Thus, the KDE project was born.

In his first public email (reproduced below) about his new 'Kool Desktop Environment' project to the de.comp.os.linux.misc Usenet group he shared his desire to use *Qt*:

"For a few weeks now a really great new widget library is available free in source and price for free software development. Check out at **www.troll.no**.

"The stuff is called '*Qt*' and is really a revolution in programming *X*. It's an almost complete, fully C++ widget-library that implements a slightly improved *Motif* look and feel, or, switchable during startup, Window 95.

"The fact that it is done by a company (Troll Tech) is IMO a great advantage. We have the resources, a superb library and they have beta testers. But they also spend their WHOLE TIME in improving the library. They also give great support. That means, *Qt* is also interesting for commercial applications. A real alternative to the terrible *Motif* :) But the greatest pro for *Qt* is the way how it is programmed. It's really a very easy-to-use powerfull C++-library.

"*Qt* is also portable, yet to Windows 95/NT, but you do not have to worry about that. It's very easy to use UNIX/*X* specific things in programming, so that porting to NT is hardly possible. :-)"

With the Linux and free software communities beginning to grow more and more, *Qt* was rapidly becoming a key component in a growing story that had high ambitions to take on the big boys in the proprietary software. Unfortunately though, not everyone was a fan of *Qt* with doubts surrounding how it was licensed.

Licensing challenges

When *Qt* was first released it was licensed under the FreeQt licence, up to it hitting v1.45. For some members of the Free Software community the FreeQt licence was a significant point of contention as the licence didn't allow modified versions of the *Qt* source code, which is a key freedom granted by the GNU Public License. Such critics were often seen praising the technical capabilities of *Qt*, but the licensing issue was an ethical blocker in the eyes of some.

To try to rectify the issue, version 2.0 of *Qt* shipped under a different licence, the Q Public License (QPL). While the QPL provided many of the core freedoms in a free software licence, and was considered by many a legitimate free software licence, the Free Software Foundation who created the GPL didn't feel the QPL was compatible with the GPL. *Qt* found itself back at the drawing board.

› **Lars Knoll (right) has been a significant force in leading both *Qt* and KDE.**

This licensing issue caused significant consternation between Trolltech and the KDE project. The KDE project was put under increasing pressure at the time to move away from *Qt* due to the licensing issues.

Stepping down from the **LXF** lectern for a moment, the author remembers this time vividly as he was actively involved in the KDE project and felt some of the brunt of this backlash against *Qt* as he was in the position of UK KDE representative at the time. Not a day would go by where there wasn't controversy surrounding this issue, which continued to act as a distraction from both the KDE and *Qt* teams, who wanted to focus on building great software. To many, the entire issue was quite depressing; everyone cherished freedom, but some felt these debates were in-fighting that slowed the wider Linux effort down somewhat. The author shared this view, although supported the call for a *Qt* that had greater freedom, but also one that benefited both free software and commercial app developers.

During this time two efforts started to resolve the issue. The first was the somewhat ambitious *Harmony* project that sought to build an API-compatible and fully free software equivalent of *Qt*. Second, the KDE and *Qt* projects agreed through diplomacy that *Qt* would not fall under a more restrictive licence, even if a company purchased Trolltech. This resulted in the KDE Free Qt Foundation which

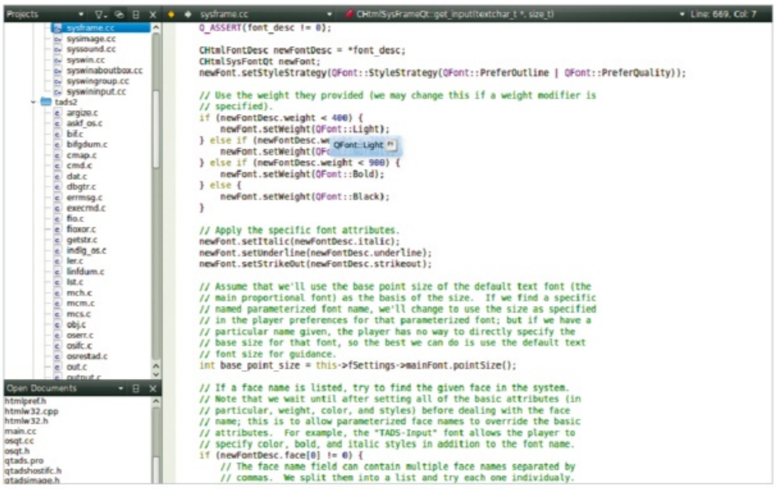
»

Governance and Community

When the *Qt* project was born it was built around a very specific set of ideas and goals from its founders, Haavard Nord and Eirik Chambe-Eng. Since those early days, *Qt* has become an increasingly open and collaboratively governed project.

Qt operates as a typical open source project. Development is done out in the open across mailing lists, IRC channels, bug trackers, wiki pages and forums. Anyone and everyone is welcome to join the project and many commercial organisations actively contribute. This includes Digia, Swedish contracting company KDAB, RIM (contributing Blackberry and QNX support), Intel (contributing *Wayland* support), and AudioCodes (contributing ClearCase support in *Qt Creator*).

Qt is also an openly governed project run under the *Qt* Project banner. Importantly, this governance is very much focused on encouraging collaboration and is based upon *WebKit*'s model of open governance.



```
Q_ASSERT(font_desc != 0);

CHtmlFontDesc newFontDesc = *font_desc;
CHtmlSysFontQt newFont;
newFont.setStyleStrategy(QFont::StyleStrategy(QFont::PreferOutline | QFont::PreferQuality));

// Use the weight they provided (we may change this if a weight modifier is
// specified).
if (newFontDesc.weight < 400) {
    newFont.setWeight(QFont::Light);
} else if (newFontDesc.we
    newFont.setWeight(QFo
} else if (newFontDesc.weight < 900) {
    newFont.setWeight(QFont::Bold);
} else {
    newFont.setWeight(QFont::Black);
}

// Apply the specific font attributes.
newFont.setItalic(newFontDesc.italic);
newFont.setUnderline(newFontDesc.underline);
newFont.setStrikeOut(newFontDesc.strikeout);

// Assume that we'll use the base point size of the default text font (the
// main proportional font) as the basis of the size.  If we find a specific
// named parameterized font name, we'll change to use the size as specified
// in the player preferences for that parameterized font; but if we have a
// particular name given, the player has no way to directly specify the
// base size for that font, so the best we can do is use the default text
// font size for guidance.
int base_point_size = this->fSettings->mainFont.pointSize();

// If a face name is listed, try to find the given face in the system.
// Note that we wait until after setting all of the basic attributes (in
// particular, weight, color, and styles) before dealing with the face
// name; this is to allow parameterized face names to override the basic
// attributes.  For example, the "TADS-Input" font allows the player to
// specify color, bold, and italic styles in addition to the font name.
if (newFontDesc.face[0] != 0) {
    // The face name field can contain multiple face names separated by
    // commas.  We split them into a list and try each one individualy.
```

› *Qt Creator* provides everything you need to start building *Qt* apps, all in one place.

guaranteed that *Qt* would fall under a BSD-style licence should no free/open source version of *Qt* be released during 12 months.

Fortunately, as we clocked into the new millennium, and as KDE was ramping up as a compelling desktop environment for Unix and Linux systems, Trolltech decided to license *Qt* under the GPL v2.0 thus ending the controversy, at least in terms of the KDE project's dependence on Qt.

There was a problem though: one of the most compelling attributes of *Qt* was it's cross-platform capabilities, but the Windows port of *Qt* was still not available under the GPL v2.0. Although many involved in the licensing debate were satisfied that Trolltech had provided a Linux port for free and made money on Windows licences, there were some who were still dissatisfied that the portability of *Qt* was still stifled by licensing restrictions.

In 2002, members of the KDE on *Cygwin* project took a stab at resolving this issue and began porting the GPL licensed *Qt* for *X Window System* codebase to Windows. This ambitious project achieved generally decent results, although it was shelved when Trolltech released *Qt4* for Windows under a GPL license in 2005. Now the entire *Qt* platform was available under a free software licence so open source apps could be easily created and distributed.

Commercial break

Traditionally the *Qt* project had always had a very clear leader and commercial backer in Trolltech. The company was founded by the founders of *Qt* and went on to hire many of the best-known *Qt* developers and engineers in the community. For those following the KDE project closely, Trolltech became a veritable vacuum cleaner that sucked up most of the talent to work on making *Qt* better, and thus providing a better set of tools for building KDE.

Although Trolltech had traditionally focused on the desktop market across Linux, Windows and Mac compatibility, the advent of the Linux kernel being able to power mobile devices was becoming more and more compelling. This revolution was in large part fuelled by the iPaq device by HP that ran Linux and a sub-set of applications. Although the apps ran technically, many of them looked like desktop apps running on a small screen. We needed a graphical toolkit optimised for mobile screens… hmmm, I wonder who could provide that?

Long before iOS and Android dominated the mobile market, Trolltech saw the opportunity for *Qt* on mobile and started working on support for *Qt* running on mobile devices. This work resulted in *Qt* Embedded; a powerful *Qt* distribution for writing mobile apps. *Qt* Embedded was effectively the same as any other *Qt* port, but ran better on lower specification hardware, was optimised for touch and provided device-level features. It was also a pretty solid start.

Device deals

A key stepping stone in this work was a deal struck between Trolltech and Sharp to release a Linux powered mobile device called the Sharp Zaurus. The device was not a phone or tablet, but a PDA (this was around the time Palm and Handspring devices were very popular), but was one of the first devices in the world to natively run Linux and certainly the first to be powered by *Qt*.

Despite enthusiastic developer interest (and the author's interest) the Zaurus was a pretty monumental flop. The limited specification and sales, and complexity of deploying applications and sharing them with others, limited the opportunities of the device, but the opportunities for *Qt* in a mobile setting were still very much present.

This was all noticed by Nokia, which announced in January 2008 its acquisition of Trolltech. At the time Nokia were getting somewhat crushed in the mobile space, with Apple dominating the market, and Android fuelling new hardware giants such as HTC and Samsung. Nokia needed to do something and identified *Qt* as a means in which they could build a powerful developer strategy.

Unfortunately, things didn't work out quite as well as expected. While Nokia put out some of its fabled 'Internet tablets' such as the N900, ultimately the devices suffered the same fate as the Sharp Zaurus by not connecting with consumers and facing hardware and feature limitations.

What followed was a tense few years for the *Q*t team and its fans. While Nokia were throwing large amounts of money to see if its products would stick, limited commercial success was putting Nokia in a tough spot. This all came to a head when Nokia CEO Stephen Elop announced a strategic partnership between Nokia and Microsoft.

This author remembers this time vividly as his inbox started filling up with queries from an unsettling number of Nokia staffers enquiring about jumping ship and coming to work at Canonical. Times were tense and many were concerned about the future of *Qt*. It seemed clear that the new Nokia Microsoft strategy didn't include *Qt* in the line-up.

Write your first Qt app

If you want to get started building your first *Qt* app, be sure to check out the whole host of resources that are available online before you get started:

- **Documentation – http://qt-project.org/doc**
- **Support Forums – http://qt-project.org/forums**
- **Regional Groups – http://qt-project.org/groups**

If you want to contribute to Ubuntu's multi-platform strategy you can also write a convergent QML app (QML is part of *Qt*) that runs across desktops, phones, and tablets. This method uses QML as a core technology plus some specialised Ubuntu components. You can find out more about development at **http://developer.ubuntu.com**.

Not only that, but for many hardened open source fans, a deal with Microsoft was simply a bridge too far.

Fortunately, tensions were relieved in August 2012 when Digia, a long-time investor in the *Qt* project and provider of consultancy services, announced the acquisition of *Qt* from Nokia. The acquisition would mean that the primary development team would become employees of Digia.

A new age of opportunity

For many the history of *Qt* seemed to be on a somewhat downward spiral up until 2012, because of the complex relationship with Nokia, lower commercial investment in the project, fewer and fewer open source developers deciding to build applications with *Qt*. The majority of Linux distributions were also beginning to ship *GTK*-based desktops and development stacks.

Since 2012 though, *Qt* has had something of a new lease of life. The project now has a new commercial backer in Digia, a trusted and long-time friend of the *Qt* project, and the project has seen significant development in *its* core feature set and capabilities (*see What on Earth is QT5, p54*).

This has resulted in a number of organisations and projects staking their technological visions on *Qt*. Examples of this include Blackberry, Jolla, and Ubuntu.

Ubuntu has built its full convergence strategy on *Qt* with not only Unity (written in *Qt*) running across desktops, phones, tablets and TVs. Ubuntu's application development is built on *Qt*, too. In fact, if you download the Ubuntu SDK you will see the entire SDK runs from within *Qt Creator*, a core piece of the *Qt* project.

More and more open source and commercial developers are now exploring *Qt* as a powerful platform to build their apps. This includes organisations and applications such as Autodesk Maya, Mathematica, European Space Agency, Dreamworks, DraftSight, *Google Earth*, HP Virtual Rooms, Lucasfilm, The Foundry's, Panasonic, Philips, Samsung, Siemens, *Skype*, *Nuke*, *VirtualBox*, *VLC media player*, Volvo, and Walt Disney Animation Studios.

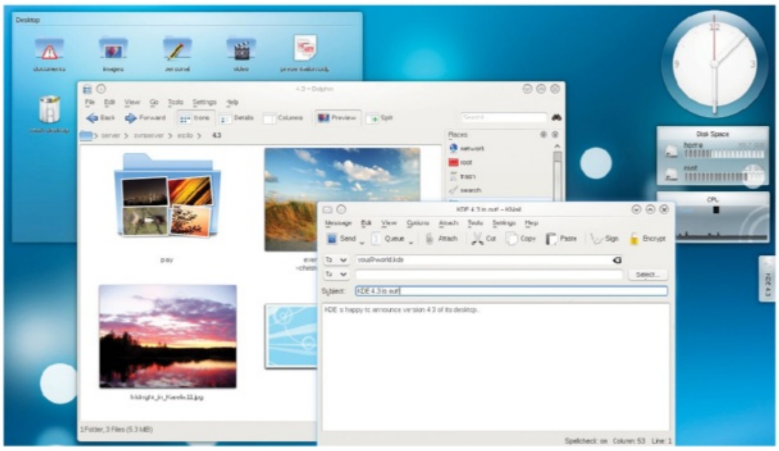
The technology

Much of the reason for this growth is that, although the history of *Qt* has seen some turbulent times, the quality, focus and innovation of the project has remained doggedly motivated and consistent. *Qt* has long benefited from tremendous leadership, and Lars Knoll is one of the most critical people in guiding the ship forward. Lars has kept the project focused, ambitious and working to resolve the challenges it has faced over the years.

The first few versions of *Qt* were originally only available for *X Window Systems* and Microsoft Windows, with the latter being available under a proprietary licence. With the release of *Qt3*, towards the end of 2001, Mac OS X was added as a supported platform.

Qt has since gone on to support Android, iOS, Embedded platforms, *Mir*, *Wayland*, QNX/Blackberry 10, vxWorks, and Windows CE. Unofficially, various communities have also ported *Qt* to OpenSolaris, Haiku, OS/2, webOS, Amazon Kindle DX and AmigaOS. Nokia also provided a Symbian port but that has since been discontinued. The quality of the codebase as kept *Qt* growing across multiple platforms.

Qt originally started out life as simply a collection of graphical components with some convenience classes that were all usable from C++. *Qt* provided these core classes and their MetaObject Compiler that would interpret custom C++ macros in *Qt* and generate C++ to provide feature unavailable in C++ at the time, such as signals and slots, introspection and asynchronous function calls.



The popular KDE environment is built using *Qt* and QML.

In the early days of *Qt* each new release would simply add to this library of graphical components and classes. As the collection started reaching a comprehensive level of features, the project then started focusing more and more on the tooling side of *Qt*.

Qt tools

One of the first major contributions here was *Qt Designer*, a graphical tool that made laying out user interfaces quick and simple. As time went on more and more of these tools would become a part of the core *Qt* distribution, such as *Qt Linguist*, Tulip, *Arthur*, *Scribe*, MainWindow, QML, and QSA.

Many of these tools were ultimately pulled together into *Qt Creator*, a full IDE with an integrated designer, debugger, source control, and other features. *Qt Creator* was a key piece in the history of *Qt*: as competing app development frameworks (such as Apple with XCode and Google with Android/Eclipse) were building end-to-end IDE/Platform/Debugger stacks, the *Qt* project was striving to produce something, which was just as compelling. *Qt Creater* effectively pulled together all of these different components into a consistent platform.

As you can see, the *Qt* project has had at times a complex history, but the dream of the founders and the many developers who helped to keep that dream alive of a simple, powerful, portable development toolkit, have remained steadfast in their vision. Where there have been challenges the *Qt* project has listened, and where there have been opportunities the project has followed them.

Qt is now in a very positive place with a strong and positive future. Significant organisations are delivering *Qt* solutions and development stacks, Digia has taken over as a capable commercial stakeholder, and *Qt* continues to grow from strength to strength. I for one can't wait to see what the next chapter in the *Qt* history books looks like! LXF

Find out more

If you want to learn more about the backers of *Qt*, Digia, the *Qt* project and how to connect with *Qt*'s community be sure to check out the following resources:

- **Digia Qt Page – http://qt.digia.com**
- **Qt Project Homepage - http://qt-project.org**
- **Qt Blog – http://blog.qt.digia.com**
- **Qt Download – http://qt-project.org/downloads**
- **Qt Community – http://planet.qt-project.org**

"Encouraging more diversity within a growing community is a strength of Python."

Photos Alan O'Donohoe

PyCon UK 2013

Les Pounder joined the annual Python gathering in Coventry for four days of community fun

Python, the well-known programming language created by Guido Van Rossum in 1991, has a large base of users and developers, and with it many Python conferences (or PyCons) dotted around the world. It's at these events that Pythonistas (those that use and love Python) come together to talk about their favourite language, how they use it and what new ideas they have.

PyCon UK was a sell-out this year with around 300 developers gathering for a four-day event at the University of Coventry's Techno Centre The conference is held over a weekend, with a large percentage of that time dedicated to prescheduled talks. There are also lightning talks, organised for the Sunday afternoon. On the last day there are a number of sprint events where developers get together to work on a project. This year Django web development was a popular sprint topic.

The sheer scale of the event is wonderful, with delegates travelling from across Europe to take part and the quality of the talks was exceptional, with experts providing an insight into their specialist areas. There were plenty of moments, where I learnt new skills and formed new ideas. Most notable for me was the DevOps based talks where I learnt how the role of development and operations is rapidly changing to meet the needs of smaller, more agile teams and how Python is being used to create an efficient workflow.

> "PyCon UK 2013 was a sell-out this year with around 300 developers."

Python is used heavily in the Linux community, particularly in development and system administration but that's far from the only areas and some of them covered at the event were a surprise to me.

PyGame is a well-known library/framework for creating 2D games using Python. But did you know that there's also a 3D library called *Pyglet* that can help you create your own version of *Minecraft* in less than 500 lines of code? Python is a versatile tool, that encompasses many roles in the Linux community, and PyCon UK is a great showcase, with plenty of opportunities to get hands on, learn from experts in their fields.

PyCon UK, also has a great social element to the conference. With delegates being treated to many opportunities to meet and mingle outside of the scheduled track of talks and workshops. On the first night, we were provided with food and drinks at a nearby venue. On the second evening, there was a lavish three-course meal provided at the university. With seating arrangements decided by the Python random library, which made the evening very fun. The pub chosen for the event was the White Friar, a cosy real ale pub and it was heavily used over the course of the event, as it is well known that Pythonistas love real ale.

The PyCon UK events, like most conferences, require lots of planning and a dedicated team to ensure success, so I took the opportunity to interview one of the organisers, Nicholas Tollervey. He spoke to me about the development of PyCon UK and, particularly, the Education Track.

› It's not a bad thing when your biggest headache... › ...is organising all the volunteers that want to help.

Les Pounder: How did the idea for PyCon UK come about?
Nicholas Tollervey: Back in 2006 the Python conferences that were available to us were too expensive, formal and commercial in outlook. There was nothing for the grassroots community. We wondered if we could organise a weekend conference that cost only £100 all in. If so, we'd be on to a winner. Five years later, we're still here and as popular as ever.

LP: What is the goal for PyCon UK?
NT: A great value conference by the community, for the community and of the community (where the community encompasses pros, corporate and hobbyist Pythonistas).

LP: Python is used in a lot of different projects: from simple coding in schools to the film industry for special effects. Can you explain why Python gets picked so much for projects?
NT: Python is well known for being easy to read and maintain, comes with 'batteries included' so you're able to do many, many, things quickly right out of the box and, as events like Pycon UK and efforts by the PSF (Python Software Foundation) demonstrate, is well known for its inclusive, fun and helpful community.

LP: With the success of the Raspberry Pi over the last 18 months has there been a surge of new developers/coders, attending PyCon UK?
NT: We sold out this year. That speaks for itself. What's more interesting is that our attendees are a more diverse lot in terms of both gender and, obviously, age. Encouraging such diversity within a growing community is a strength of Python.

LP: This year, you repeated the 2012 event's education track, what was the idea behind this track and what challenges did you face in merging a developer-centric conference with this aspect of the conference?
NT: The philosophy is simple: if you bring a diverse group of people together who are passionate about a shared topic then interesting things are bound to happen. Last year was a great success. Some of the projects relating to Python's logo-like turtle module that were featured in this year's main conference were directly inspired by last year's collaboration with teachers.

> "There's a lot of latent enthusiasm for education, learning and teaching."

Another example of 'interesting things' happening is Alan O'Donohoe [founder of Raspberry Jam]. He has been attending Pycon UK for three years now and I've watched with delight as he has flourished along with his Raspberry Jam phenomenon. It was only natural that we should collaborate like this for the kid's track since he's such an expert. It was a pleasure seeing him in action. Perhaps the biggest challenge for us was cat-herding the large number of developers who volunteered their time and effort in the education track. There's a lot of latent enthusiasm for education, learning and teaching in the Python community. I »

Teach kids to code

Teaching the next generation of programmers was the big topic at PyCon UK. Teachers need your help to make this happen.

In recent issues of *Linux Format*, [**LXF176**, Organise Your Own Event, for instance, p44] you will have seen me write about Code Club, STEM and Young Rewired State. These are all great causes, and I would urge you sign up.

In the 1980s the UK had a massive 'bedroom programmer' scene, where children taught themselves to code, and with the Raspberry Pi and Python, we are seeing a resurgence of this scene. But schools around the UK haven't got the skills necessary to teach computing, and would love to have *you* involved in their lessons, leading sessions and advising the school on how to teach computing.

Get in contact with your local school, and make an appointment to speak to their head of ICT and ask how you can help. If the school doesn't have a Code Club, perhaps you will be the one to start it? Subject to the usual screenings, of course.

Code Club, starts with Scratch to help club members understand the concepts of syntax and over the course of a few terms, the class will progress to basic HTML and CSS. Python can be introduced after the HTML term and you are free to develop lesson plans to continue the classes coding lessons.

guess the trick is working out how to direct this to as good an outcome as possible for developers, teachers and students.

LP: From the Education track, what was your stand out moment and why?
It has to be the kids and their infectious enthusiasm. I was especially proud that we had a 50/50 split in terms of gender. This bodes well for the future.

LP: This year, in the Education track, you had Martin O'Hanlon talk about using Minecraft for the Raspberry Pi and it's related API, written using Python. Are there any other Python libraries, that would allow teachers and children to quickly develop games/apps?
NT: Mojang has done a great thing making *MinecraftPi* freely available to program. It's basically digital Lego. That it's perhaps the world's most popular game at the moment and kids get immediate feedback via the Python API means we have the potential to inspire lots of young programmers with a very powerful development environment.

For those who are a bit older, PyGame is well known and relatively easy to get into. For example, I managed to create a Pygame based Christmas card in about an afternoon without having any previous knowledge of the Pygame API (**https://github.com/ntoll/PyMerryChristmas**).

LP: Will you have the Education track for 2014, and if so, what would you like to do to make it bigger?
NT: Yes we're definitely holding another education track in 2014. We'd love more collaboration with teachers. They're the key to inspiring and educating the next generation of programmers and innovators here in the UK. If you are a teacher or know a teacher who is interested in programming, tell them to come along next year! Evidence suggests that the teachers who turned up this year had a lot of fun and found the experience very rewarding (watch the promotional video here **http://youtu.be/qOya5R9g1T8**).

There's even time to watch the XFactor results.

The Education track had a 50/50 gender split.

LP: For anyone interested in trying Python, what would be the best resource for their reference?
NT: There are lots of places to find out about Python. There are several user groups dotted around the country. You could take part in online courses such as Codecademy (**www.codecademy.com/tracks/python**) or even go all 20th century and actually read a book! There's also the wider Python community (**www.python.org/community**).

Education track

PyCon UK first introduced the Education track to its conference in 2012, which sought to bring teachers and developers together to learn more about each others' roles, experiences and to help solve any issues that each may have faced. This track in 2013 was, by far, the most popular aspect of PyCon UK and provided lots of great moments, such as controlling a drone by using *Minecraft*, a Raspberry Pi and just a few lines of Python.

Schools are doing more with code and exposure to Python programming in schools has significantly increased in the last 18 months, largely thanks to the Raspberry Pi. In 2014, changes to the curriculum for 11- to 16-year-old children will also mean teachers have a greater amount of content to learn before they can teach the new curriculum. And with the curriculum changing rapidly, it appears that the goal posts are constantly moving. Teachers, of all key stages, are understandably nervous of the change. The 'Microsoft way' of teaching IT has prevailed for a long time and many of the old guard have extremely limited knowledge of coding.

Children are now expected to learn about algorithms and syntax from an early age, and this is where Python is such a key product. With Python's use of indentation and white space, it clearly illustrates both of these elements on the screen for both the teacher and pupil.

Our pick of Python resources

OCR Resources – The OCR is one of the examining bodies in education (Oxford, Cambridge and RSA Examinations), have produced a series of lesson plans, which are rooted in Python and the Raspberry Pi, you can find out more about their fantastic resources here. **http://bit.ly/1aFzfTB**

Simon Haughton's Python guide – Simon, a teacher, has produced a wonderfully in-depth set of free resources, which is invaluable for teaching Python to children at primary school level.
http://bit.ly/17I9LUI

PyGame – Have you ever wanted to make a 2D game? Well take a look at PyGame, this library of resources provides everything that you need to build simple games quickly.
http://www.pygame.org/news.html

PiFace – This is a little advanced, but PiFace is a great add-on board for the Raspberry Pi. This will enable you to quickly attach all manner of components to the Raspberry Pi, including devices which use more than 5V of power, meaning that Raspberry Pi powered nerf sentry gun is almost ready for combat.
http://pi.cs.man.ac.uk/interface.htm

MotoPiTX – Have you dreamt of building your very own robot? Well, with this your dreams can come true. MotoPiTX was a successful Kickstarter campaign, and is wonderfully simple to use kit. Within a few hours, you can have a sophisticated robot to do your bidding.
https://github.com/Boeeerb/MotorPiTX

Code Academy – OK, this isn't a Python exclusive resource, but it is a great place to show how easy Python is, all from the comfort of your web browser.
www.codecademy.com

Nicholas Tollervey doing his session.

Python-controlled ARs!

During the weekend, I spoke to Vikki Dodd, who is the Assistant Principal of ICT at Our Lady's Catholic High School, Preston and she explained the current state of ICT in secondary schools.

Les Pounder: Let's start by talking about the changes to the ICT curriculum from 2014.

Vikki Dodd: From 2014, children will be taught computing from age five onwards. Key stage 1 is for children aged five to seven years and in this key stage they will be taught basic algorithms, which will form the basis of their first programs. In key stage 2, which covers children aged 7 to 11, children will learn to create basic programs, which ultimately will control small robots. In key stage two, teachers need to know more about algorithms, logic, selection and iteration. Children at key stage 3, aged from 11 years, need to understand a visual language, such as Scratch, and a text based language. We are pushing for Python to be the main contender, as we feel that it has the right balance of ease of use and powerful features.

LP: What's the typical reaction to teaching computing?

VD: The majority of teachers, at primary school level, have little or no qualifications in computing. At primary level, teachers are largely required to be generalists, covering many subjects. Words such as algorithms, logic, iteration are quite scary to them, as they have never come across them before, from the point of view of coding and logic. Currently, what children are learning in key stage 4 (GCSE level), such as functions, is being pushed down to key stage 3, meaning that teachers need to understand these complex concepts and convey them to their classes a lot sooner than previously.

LP: How can teachers gain the confidence to teach computing to their classes?

VD: Teachers, typically receive training in the form of Continuing Professional Development (CPD). This is where their department receives a budget to take part in specialist training courses. Generally, this budget is quite low, and is spread across Computing and Business Studies. To help teachers learn more about computing, it would be great if developers contacted their local schools, and offered to volunteer their time, to help us all understand what we need to do to help children learn computing.

A great way to do this, would be via a Code Club (**www.codeclub.org.uk**), and I would love to see more people volunteering to lead a club, and by registering as a club volunteer. You can also register as a STEM (Science Technology, Engineering and Maths) Ambassador.

LP: What can developers/coders do to help teach Python?

VD: Finding your local Computing At School hub (**www.computingatschool.org.uk**) would be a big step, I would suggest that anyone interested, signs up via the website. Generally at a meeting there are lots of teachers, who are looking to swap skills, and learn more about programming and computing in general. Anyone with Python skills will be a big asset to the group.

> "Python's flexible approach is paramount to its success in education."

Python in the classroom

PyCon UK shows that Python's flexible approach is paramount to its success in education. From learning the basic structure of a program, children can quickly adapt and devise their own programs. A great example of which was Martin O'Hanlon's fantastic *Minecraft* session. The Python API enabled new coders to quickly build in-game content and using *Minecraft* to teach programming is a fantastic example of using play to teach in a classroom setting. It produces fantastic results kids can see – and with children, immediate results often produce the best responses. LXF

Lesson ideas

Python has truly changed the way that computing is being taught in schools across the UK. Let's take a look at a handful of lesson ideas for the classroom or home.

Python-powered traffic lights

Using the Raspberry Pi GPIO connections, a red, amber and green LED, some resistors wires and breadboard, you can easily program your own set of working traffic lights. This activity is a great introduction to Python, and produces quick results, which will keep a child engaged.
http://bit.ly/1cWpXq3

Minecraft & Python

Martin O'Hanlon has produced a fantastic resource, which uses *Minecraft Pi Edition*, and a Python based API. With it you can quickly create new landscapes, buildings and messages in the *Minecraft* world. Martin's website contains lots of examples and videos.
http://bit.ly/1idi4Nu

Scratch and the GPIO

Scratch comes pre-installed with Raspbian, on every Raspberry Pi. But there's another version of Scratch, which uses an underlying Python script to enable Scratch to talk to components attached to the GPIO. Motors, servos and LEDs can all be used via Scratch. Simon Walters has produced a fantastic guide and he has even built a Scratch controlled, maze solving robot.
http://bit.ly/1a0dkpa

Make your own Minecraft

This is a little more advanced than the previous lesson ideas. But did you know, that using less than 500 lines of Python, and a 3D game library, you can make your own version of *Minecraft*?
http://slidesha.re/17ffas9

What on Earth is Qt5

Marco Fioretti delves beneath the glossy exterior of the latest release.

Q So fill us in on what *Qt5* is? I do hope the name is a hilarious play on words and not just a boring old abbreviation!

A *Qt5* is the latest version of the *Qt* application framework, designed to help developers create applications and graphical user interfaces quickly and easily across a number of platforms. Its name stands for Quasar Toolkit, and is sort of funny, I guess, as when you say *Qt* out loud it sounds like 'cutie'. Although, according to the official website it's supposed to be pronounced 'cute'.

Q Ah, so not that hilarious then. You mentioned that it can be used across platforms – how flexible is it?

A When Digia, the developers of *Qt*, and the *Qt* Project describe *Qt5* as cross platform, they're not kidding. As well as Linux, you can create applications for Windows, Mac, Android, iOS, Blackberry, QNX, Sailfish, Solaris and Integrity.

Q Wow. That's quite a few platforms. So what exactly can I do with them?

A The idea behind the *Qt* Project is to make creating good looking, robust and easy-to-use applications for any of the platforms that are supported. It gives developers access to the *Qt* C++ UI framework, letting them create sophisticated applications with good looking and user-friendly interfaces. With access to OpenGL/OpenGL ES graphics acceleration technology, particle systems and shader effects, along with *Qt* Multimedia and *Qt* Graphical Effects tools, developers can make some seriously good-looking applications.

Q I can sort of understand the 'cute' moniker a little more then. So what's new with version 5?

A The aforementioned graphical tools are one of the big new features of *Qt5*. It uses an OpenGL-based scene graph to boost the graphics of *Qt* Quick, resulting in some beautiful animations, graphical effects and particle systems, even if the hardware isn't particularly powerful. This is especially useful for making good looking apps that are developed to run on mobile devices.

Q Sounds great, but are there any examples of *Qt5*'s graphical prowess that I can look at?

A There are indeed. Head over to **http://qt-project.org/videos/watch/livecoding-video-effects-with-qt5** to see an example of 720p livecoding video effects in *Qt5* and witness how quick and easy it is to add and configure video effects on the fly. On the *Qt* Blog (**http://blog.qt.digia.com**) there's an excellent demonstration of applying shader effects to a video using *Qt* Quick 2, which forms a core of *Qt5*. You can see the demonstration at **http://blog.qt.digia.com/blog/2012/02/29/pimp-my-video-shader-effects-and-multimedia**.

Q So *Qt* Quick 2 is a big part of *Qt5*? What are the new features does it bring to the table?

A *Qt* Quick is the main library for writing QML applications, with a particular focus on creating user interfaces. The latest version of *Qt* Quick, version 2.0, drops the QDeclarative classes of *Qt* Quick 1 in favour of a new set of C++ classes. Along with the advanced animation and graphic tools we talked about earlier, a built-in Scene Graph rendering system and Canvas API are included to make creating gorgeous user interfaces as easy as possible for developers, no matter what platform they are targeting. All important JavaScript engine optimisations for QML are also included. Pop over to **http://qt-project.org/videos/watch/whats-new-in-qtquick-2.0** to see a video that highlights some of *Qt* Quick 2's best new features.

> **"Create good looking, robust and easy-to-use apps for any of the platforms supported."**

Q OK, there are a lot of abbreviations being thrown around here – and none with even remotely humorous names anymore. What is QML?

A QML stands for either *Qt* Meta Language or Qt Modeling Language, depending on who you ask. Apart from arguing about what the letters stand for – instead of the pronunciation – the creators of QML designed this JavaScript-based language for creating applications that are user interface-centric.

Because it's a central component of *Qt* Quick, which used to be developed by Nokia, there's an inevitable focus on mobile apps, especially touchscreen enabled ones, where ease of use for a user interface (as well as being fast and light) is a priority.

QML documents are made up of various QML elements that offer powerful tools for creating and controlling the graphical and behavioural properties of objects. When combined these elements can create components with a large variety of complexity from simple buttons to complete programs with internet connectivity. Also, as if QML (and by extension *Qt* Quick and *Qt5*) wasn't flexible enough, QML can be augmented with standard JavaScript, as well as integrated with C++ components as well.

Q Wow, so flexibility seems to be a key word here. One word you've just used, however, worries me: "Nokia". What does nobodies favourite on-the-verge-of-irrelevancy mobile phone company have to do with *Qt5*?

A Thankfully, not much any more. Nokia's acquisition of Trolltech, the original producer of *Qt*, occurred in 2008 where it paid around €104 million and renamed the company *Qt* Development Frameworks. The idea was that this would lead to a renaissance of apps on Nokia's struggling Symbian mobile operating system, thanks to the easy to use tools included with *Qt*. However only a few years later Nokia finally put Symbian out of its misery and dumped it in a shallow grave next to the decomposing corpses of WebOS and Palm OS, and sold *Qt* Development Frameworks to fellow Finnish company Digia, who continue to develop Qt. (*See Inside Qt on page 46 for more*).

Q Ah right. So, I suppose Nokia then went on to use a successful and world-conquering new mobile operating system?

A Um, not exactly. It now creates handsets exclusively for Windows Phone 8. While Android and iOS are supported in *Qt5*, Windows Phone remains unloved. Some rather enterprising developers have been looking at ways to port their *Qt* apps over to Windows Phone, navigating some of the more OpenGLphobic (it's a word) barriers in the mobile OS. However, until Microsoft drops its 'DirectX-way or the highway' attitude, porting *Qt* apps to Windows Phone will remain difficult process, meaning that the operating system that's in need of killer apps continues to miss out on a lot of the popular apps that appear on other platforms.

Q Does that mean that Qt5 is all about mobile apps?

A Not at all. Internet-based apps are well looked after with the *Qt* WebKit, which is a web content rendering engine. It's based on the open source WebKit project, allowing developers using *Qt5* to support a broad range of web technologies and standards. The *Qt* WebKit brings HTML5 and all of its advancements to *Qt5*, including CSS filters, WebGL, Canvas support and HTML video. *Qt* WebKit is one of the many modules included in *Qt5* that offer developers the tools to create some really cool apps, no matter what platform they're running on.

Q What other modules are included in *Qt5*, and what do they do?

A There are two main types of modules in *Qt5*: *Qt* Essentials and *Qt* Add-ons. *Qt* Essentials modules are general purpose, and useful for the majority of *Qt* applications. These modules include *Qt* Core, which all other *Qt* modules rely on, *Qt* GUI, which includes classes for graphical user interface components, *Qt* Multimedia, which handles audio, video, radio and camera features and the aforementioned *Qt* WebKit, amongst many other things.

The *Qt* Add-ons modules, on the other hand, are for much more specific purposes. These can involve handling compatibility between languages for various platforms, such as the Active *Qt* module, which is specifically for Windows platforms, and includes classes for applications the use ActiveX and COM. Meanwhile, the *Qt* D-Bus module handles inter-process communications over the D-Bus protocol on Unix platforms. *Qt* SVG contains classes for displaying the contents of SVG files, while *Qt* Script and *Qt* Script Tools provide classes for making *Qt* applications scriptable.

Other Add-on modules include *Qt* Declarative, which deals with *Qt4* compatibility, and *Qt* Print Support, which as the name suggests lets developers include printing functionality to their applications. *Qt* Graphical Effects, *Qt* Image Formats and *Qt* OpenGL handles the more glamorous side of things.

Q So what apps have been built in *Qt* and with *Qt5*? Are there any big ones that I'll have heard of?

A Some of the better known applications that have been built in the *Qt* language, or use elements of it include; *Google Earth*, *VLC Media Player*, *Guitar Pro*, *Scribus*, Last.fm, Launchy and Spotify. Even some games such as *City of Heroes* have used elements of the *Qt* language.

If you point your browser towards **http://qt-apps.org** you will see a list of *Qt* apps that are currently in development. Ignore the comments at the bottom of the page though, unfortunately its full of spam. Of course, just because they're written in *Qt* doesn't mean they've used *Qt5*. However, as more *Qt* developers upgrade to *Qt5* we should soon start seeing improved – or completely new – apps that take full advantage of all the new features of *Qt5*. LXF

Dr Brown's Administeria

Dr Chris Brown
The Doctor provides Linux training, authoring and consultancy. He finds his PhD in particle physics to be of no help in this work at all.

Esoteric system administration goodness from the impenetrable bowels of the server room.

The Turing test

It was Douglas Adams who invented the Electric Monk, a labour-saving device that believed things for you, saving you the bother of believing them yourself.

In a similar spirit, I've invented the Electric Tweeter, an app that tweets on your behalf. Intending to present this as a tongue-in-cheek invention, I discovered that there is a software product called AutoTweeter (**www.autotweeter.in**) that does this – sort of. You have to enter your tweets into a spreadsheet and the program handles the mechanics of sending them. Not to worry, though – for a few dollars you can buy thousands of pre-written tweets.

What I had in mind was a more sophisticated app that would analyse your previous tweets, monitor your emails and browsing history, and originate the tweet on your behalf. And it occurred to me to ask – how hard would it be to pass the Turing test for tweets? (The Turing test is a test of a machine's ability to exhibit intelligent behaviour indistinguishable from that of a human. A human and a computer interact via text messages. A human observer monitors the interaction. The computer passes the Turing test if the observer can't reliably distinguish which is which).

In the meantime, I'm writing an Electric Tweet Reader, a little program that relies heavily on **/dev/null** and reads your Tweets for you. Once these two apps are in use, their bandwidth requirements will spiral out of control and the internet will explode in a blaze of heat and light, allowing us to return to having a life.

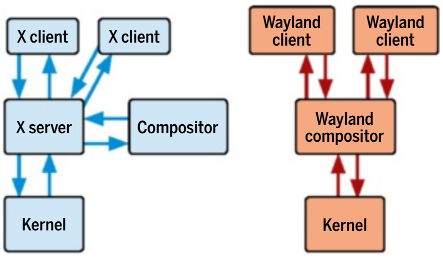
chris.linuxformat@gmail.com

What's a compositor?

Carefully sidestepping the *Mir/ Wayland* argument, Dr Brown answers this burning question.

The *Mir/ Wayland* debate got a shot of media attention lately when Intel announced that it wouldn't accept a patch to the Xorg video driver to support *Mir*. It was a slap in the face for Canonical. My reason for mentioning this is not to get drawn into the debate. *Mir* and *Wayland* are both compositing display servers, and I realised that I didn't entirely understand what a compositor does. So I thought it was time to find out…

In a traditional X windows environment, each graphical program (X client) is responsible for drawing its own window on the screen, and for re-painting the window (or parts of it) when an overlying window is moved or closed. But when using a compositor, each app maintains its window in a buffer, and the compositor is responsible for combining the windows from all the apps onto the screen.



› **Current systems run a compositor alongside X, but *Wayland* and *Mir* can dispense with the X server altogether.**

The compositor can make the window semi-transparent, scale and rotate it, project it onto the faces of a rotating cube, deform it like a soap film, slide a digital magnifying glass over it, or make it grow out of an icon on a task bar like a genie from a lamp. In all these cases, the app that is drawing the window is unaware that these transformations are happening.

We've had compositors around for some time – *compiz* is a popular example, as are *Xfwm* and *Enlightenment*. Mac OS X has *Quartz*. But in current implementations, the compositor sits atop X windows – the mechanism that Linux (and UNIX before it) has used for donkey's years to drive bitmapped screens. Extensions to the X server protocol were added to allow access to the 3D hardware acceleration of modern graphics cards. But *Wayland* and *Mir* can potentially dispense with X altogether – a shock to the system for old hands, but a good thing if you want a lighter-weight implementation for tablets and phones.

The *Wayland* website has a good discussion about the limitations of a separate 'X Server + compositor' architecture, and the benefits of combining the roles of the two (**http://wayland.freedesktop.org/architecture.html**).

Compositing in film

In the film industry, compositing refers to combining separate elements into a single scene; usually foreground action against a separate background. Older (optical) techniques included back-projection, multiple exposures using mattes, and "glass shots". Newer techniques include "chroma key", where a scene colour is replaced by a background image. It's often used to generate "virtual" TV studios.

Everything is a file

Armed with a torch and wellies, the intrepid Doctor goes exploring through the filesystem to find out what kinds of creatures live there.

Run the command **ls -l** on any directory, and most of the files you'll see will be – well – just ordinary files. But there are several other critters that live in the filesystem, and this month I'm off to find them.

Regular files

Plain ol' files are by far the dominant species, and you probably have a pretty good idea what a file is. But even here there are a few things that might surprise you. For example, none of the files under ***/proc*** and ***/sys*** exist in the sense that their data is spinning around on the disk. Instead, they are entirely a figment of the kernel's imagination, and give us a file-like glimpse of internal kernel data. To make any sense of this you may need to broaden your concept of what a file is. From a programmer's point of view, a program accesses a file's contents using four system calls: **open()**, **read()**, **write()** and **close()**. If the kernel responds to a **read()** call by returning data, then as far as the programs in user-space are concerned, the thing really is a file and you can look at it with the usual looking-at-files programs, such as **cat** and **less**.

Try doing:

```
cat /proc/cpuinfo
```

and that's exactly what's happening. You can tell that there's something weird going on, because if you try:

```
ls -l /proc/cpuinfo
```

it will show zero length (clearly a lie), and its modification time will always be the current time.

Directories

You probably think of directories as containers for other files, and that's true up to a point. But files don't physically exist inside directories in the way that my coffee exists inside my mug. Directories just contain links that point to the file. It's the links that give names to files. So if, for example, I move a file called **foo** up one level in the directory tree like this:

```
$ mv foo ..
```

I'm not actually shovelling data between the two directories, I'm merely moving the link (there's one exception to this: if I move the link to a different partition, the data has to move too). Similarly, if I create an extra link to **foo**, like this:

```
$ ln foo bar
```

I'm not creating a second copy of the file, I'm just giving my file a second name. If I edit **foo**, then look at **bar**, I see the change. There's only one file. My younger son is called Matthew, though his mates call him Matt. There aren't two people, just one person with two names. If I give Matt a cup of coffee, Matthew is refreshed.

When I teach all this filesystem stuff in class, things usually go swimmingly until we get to the **ln** command. Then brows get furrowed and questions get asked. Commonly, students ask "why would you want a file to have more than one name?" Well, here's one example, again from a training environment. A shell scripting course uses a sample data file called **shopping.txt** that's used in several chapters. The exercises for each chapter have their own directory, with names like **chap1**, **chap2** and so on. Instead of having a separate copy of the **shopping** file in each directory, we have just one copy with a link in each directory that needs it. As another example, on my Ubuntu system the names ***/usr/bin/pstruct*** and ***/usr/bin/c2ph*** are links to the same program. The program (which does something mind-numbingly obscure) behaves somewhat differently, depending on which name it is invoked by.

Having said all that, I'll admit that files with more than one link are quite rare, and I wouldn't want to oversell the benefits.

Now, there are a couple of things you can't do with hard links. First, you can't create a second name for a directory:

```
$ ln /home myhome
ln: `/home': hard link not allowed for directory
```

This restriction is there for a purpose. Ultimately, it's this rule that makes sure that the filesystem structure really is a tree, rather than an arbitrary interconnected web of directories.

The second thing you can't do with hardlinks is link across partition boundaries. That is, you can't put a link into a directory in one partition that references a file in a different partition:

```
$ ln /etc/hosts myhosts
ln: creating hard link `myhosts' => `/etc/hosts': Invalid cross-device link
```

Here, I'm trying to put in a link into my home directory (which is on its own partition) to a file in the root partition. It doesn't work.

If you think about it for a moment, this limitation becomes »

Counting the files

You don't really need a torch and wellies to find these critters in the filesystem. You can easily track them down with the **find** command. To find sockets, for example:

```
# find / -type s
```

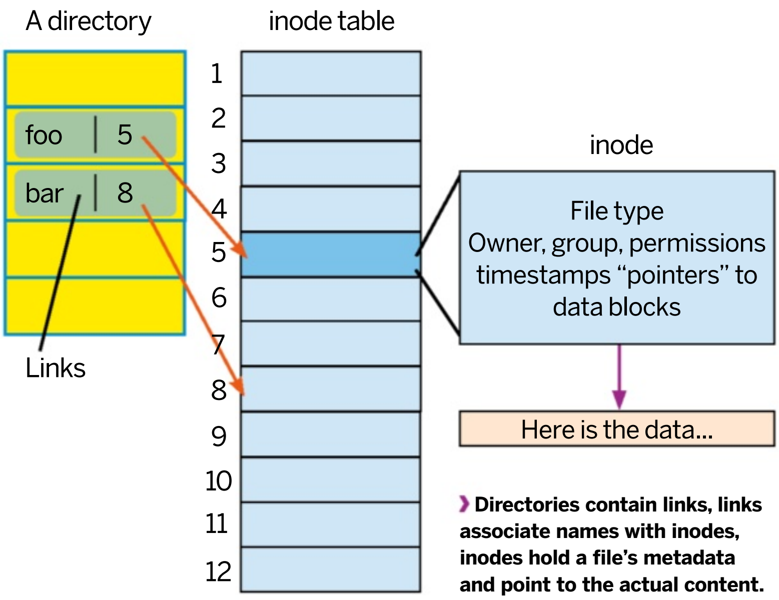
Building on that, I wrote a little shell script to generate the "frequency" figures in the table, like this:

```
for t in f d l b c p s
do
  echo -n $t :
  find / -type $t 2>/dev/null | wc -l
done
```

If you'd like to try it on your own system, run it as root, otherwise there will be quite a few directories that you can't access.

Type	Description	Frequency (Ubuntu)	Frecuency (CentOS)
-	Ordinary file (hard link)	837239	164232
d	Directory	154556	20809
l	Symbolic link	126858	21184
b	Block device	31	31
c	Character device	212	149
p	Named Pipe	0	4
s	UNIX domain socket	26	101

The seven file types found in a Linux filesystem, and their frequencies in two sample Linux distributions.



❯ **Directories contain links, links associate names with inodes, inodes hold a file's metadata and point to the actual content.**

clear. A link translates a name to an inode number, but there's no way to say "this link **foo** relates to inode no. 14328 on that other partition over there".

Which brings us to:

Symbolic links

Symbolic links (also known as symlinks or soft links) are very different. A symlink is a small file in its own right, whose content is simply another filename; usually called the "target" of the symlink.

They are a little bit like shortcuts in windows. Because of the way they work, symlinks allow you to break the rules (or if you prefer, "overcome the limitations") imposed by hard links. These two commands both work:

```
$ ln -s /etc myetc
$ ln -s /etc/hosts myhosts
```

And we can see the symlinks like this:

```
$ ls -l my*
lrwxrwxrwx. 1 chris chris  4 Sep  8 19:08 myetc -> /etc
lrwxrwxrwx. 1 chris chris 10 Sep  8 19:08 myhosts -> /etc/
hosts
```

Notice the **l** in the first character position. Notice, also, the **rwxrwxrwx** permissions. As far as I know, these permissions are entirely irrelevant – it's the permissions on the target file that matter.

This is all fine and dandy, but symbolic links do throw up a few questions. If you give a command a symlink name as an argument, should it operate on the link, or on the file it links to? The answer is "it depends", and most commands do the "right thing". For example, **rm** removes the symlink, not the target file, whereas **chmod** operates on the target file not the symlink. Some commands have a default behaviour that can be modified by a switch. For example, neither **ls** nor **tar** follow symlinks by default, but have options (**-H** and **-h** respectively) that tell them to do so.

Symlinks can cause other surprises. For example, if I change directory to the **myetc** link I created earlier:

```
$ cd myetc
```

then a subsequent **pwd** command will report:

```
$ pwd
/home/chris/mytec
```

but really we're in **/etc**. This is confusing. You also get strange behaviour if you delete the file that the symlink points to. You get a broken link. In fact, **ln -s** doesn't even check that the target file exists when you create the link in the first place. This command 'works':

```
ln -s xxx yyy
```

even though there's no file called **xxx**.

Finally, you can create havoc by creating a symlink in a directory that points to a directory higher up in the tree, creating a loop in the filesystem graph that can be, at the least, confusing.

If you're getting the impression I have reservations about symlinks, you'd be right! Used sparingly and wisely, they can do useful things that you can't do any other way. Used carelessly, they're a recipe for headaches.

Take a look at the table, which shows the actual count of each of the file types that turned up on two sample systems: my day-to-day Ubuntu installation and a bare installation of CentOS. The differences between the two sets of figures probably reflect the number of user files and software packages present in these two systems, rather than any fundamental difference in the two distributions. One thing's clear, though – regular files, directories and symlinks between them account for 99.9% (or more) of files.

I must admit I was shocked by the number of symlinks my searches turned up. Personally, I think that they are such a potential cause of confusion that I'd like to see the govermment impose a small tax on their use. Not much – maybe just a penny a link – but enough to make you ask yourself "Is my symbolic link really necessary?"

Block and character devices

It's time to move on to our remaining four, much rarer, file types. I'll discuss block and character device files together. These links serve to give names to hardware and they are almost certainly all in the **/dev** (short for devices) directory. Let's take a look at a couple of examples:

```
# cd /dev
# ls -l sda1 console
crw-------. 1 root root 5, 1 Sep  8 18:54 console
brw-rw----. 1 root disk 8, 1 Sep  8 18:54 sda1
```

Notice the **c** and **b** in the first character position. Notice, also, the two pairs of numbers in the position where the size would usually be. These are known as the major and minor device numbers and serve to identify the kernel driver that will be used to access the device. Broadly speaking,

Links and directories

If you do a long listing of a directory, you will often see a high link count; for example:

```
$ ls -ld /usr/lib
drwxr-xr-x 220 root root 49152 Sep  7 06:42 /
usr/lib
```

In view of my comments about not being able to create additional links to directories, where do these 220 links come from? Well, one of them is the link in the parent directory (**/usr** in this case). This is the link that's actually stitching the directory into the tree. Then there's the self-referential **.** link present in every directory. The other 218 are the **..** links in all the sub-directories. There's nowhere else they can come from.

a character device is one that deals in terms of a sequential byte-by-byte flow of data, whereas a block device is storage with blocks that can be randomly addressed and accessed. Disk partitions are the classic block devices, but there are sometimes other software-based block devices layered on top of these that are used, for example, for logical volumes, RAID storage, or encrypted partitions. These block devices are important because Linux needs block devices to build its filesystems on.

Not all entries in ***/dev*** represent real hardware. There are software-based 'devices' as well. Perhaps best known is ***/dev/null***, which simply absorbs anything you write to it and is useful when you want to explicitly discard a data stream. It's sometimes called the "black hole". Then there's ***/dev/zero***, which delivers an endless stream of zeros, and (less trivially) ***/dev/random*** and ***/dev/urandom***, which deliver streams of random numbers. These are all character devices.

Everything's a file, except...

The one exception to the "everything's a file" rule that I can think of is the network interfaces, conventionally called eth0, eth1, wlan0, and so on. These do not have entries in the filesystem. If you can think of any other named "thing" in Linux that doesn't have filesystem entries, then please get in touch with me at: **chris.linuxformat@gmail.com**

Named pipes

Named pipes are so rare in the Linux filesystem they should perhaps be classified as an endangered species. In fact, on my Ubuntu system they are extinct in the wild, and my CentOS system has only four. Named pipes provide a way for one process to send a stream of data to another. They don't need to pre-arrange this, but they do need to agree on the name of the pipe.

Named pipes are uni-directional. They have an upstream end that you can write to, and a downstream end that you can read from. Here's a little experiment to demonstrate. Set yourself up with two terminal windows, side by side. In the left-hand window, create the pipe:

```
$ mkfifo /tmp/mypipe
$ ls -l /tmp/mypipe
prw-rw-r-- 1 chris chris 0 Sep  9 06:45 /tmp/mypipe
```

Notice that the command to create a named pipe is **mkfifo**. (FIFO stands for "first in, first out"). Notice, also, the initial **p** in the output from **ls -l**.

In the left-hand window, run the command:

```
cat > /tmp/mypipe
```

This command will block, waiting for you to type stuff on the keyboard. Now, in the right-hand window, run the command:

```
cat < /tmp/mypipe
```

This command will also block, waiting for data to arrive in the pipe. Now start typing lines of text into the left-hand window. The text will be written to the pipe by the upstream **cat**, read by the downstream **cat**, and displayed in the right-hand window. This will keep going until you enter Ctrl+D in the left-hand window. At this point, the upstream **cat** will exit, and the downstream **cat**, noticing that the upstream process has gone away, will exit also.

There are also anonymous pipes, though they don't really belong here because they don't have an entry in the filesystem (that's why they're anonymous). For two processes to communicate through an anonymous pipe, they have to have a common parent that creates the pipe and hands the writing and reading ends down to its children. This is precisely what happens when you use a pipe on the command line.

Sockets

So, finally, we come to sockets. The sockets that live in the filesystem are sometimes called UNIX domain sockets to distinguish them from the TCP/IP domain sockets that allow programs to communicate over a TCP/IP network. UNIX domain sockets also provide inter-process communication, but they are more restrictive because the client and the server both need to be on the same machine. Unlike named pipes, sockets are bi-directional, however there is an asymmetry in how the socket is handled by the two ends of the connection.

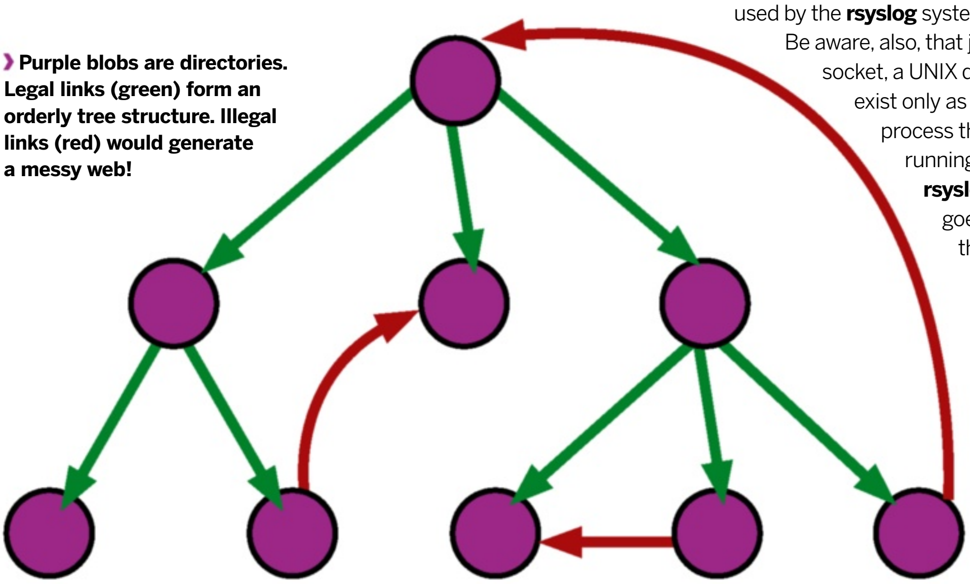
One end (the server) passively listens for connections on the socket, the other end (the client) actively connects.

It's not so straightforward to come up with a command-line demonstration of sockets, but if you want to see one in the wild try this:

```
$ ls -l /dev/log
srw-rw-rw-. 1 root root 0 Sep  8 18:55 /dev/log
```

Notice the **s** in the first character position. This socket is used by the **rsyslog** system logging daemon. Be aware, also, that just as for a TCP/IP socket, a UNIX domain socket will exist only as long as the server process that created it stays running. If you stop the **rsyslog** service, the socket goes away (if you try this, do remember to restart the daemon afterwards).

Well, something highly unusual has happened this month: I've run out of things to say and space at exactly the same time! See you next time. LXF



› Purple blobs are directories. Legal links (green) form an orderly tree structure. Illegal links (red) would generate a messy web!

LXF HotPicks

Thunderbird » XRoar » Mars Sim » CropGUI » DigiKam » Fgallery » Checkbashisms » Dust Racing 2D » Chrzazscz » bd » SeaMonkey

Mike Saunders
Has visited every single nook and cranny of the internet, and, therefore, knows where the best open source gems are hidden.

Email client

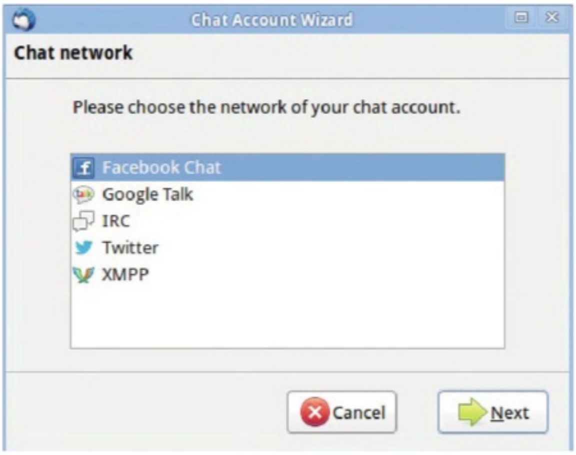
Thunderbird

Version: 24 **Web:** http://mzl.la/15NMss2

Back in the late 1990s, almost everybody used a desktop-based email client. There were various webmail offerings from Hotmail, Yahoo and co, but they were clunky, slow and desperately lacking in features compared to native applications.

The arrival of Google Mail changed everything: here was a webmail service that was fast, attractive and full of features, and (thanks to copious use of Ajax) it behaved more like a normal desktop program. Today, most of us use webmail, but there are still many good reasons to choose a native application.

The biggest benefit of a native app is offline support. Mails are stored on your computer, and you can access them whenever you need to, regardless of whether you're connected to the internet. This also helps if you're using mobile data plans: having your email stored locally means you don't need to waste bandwidth when reading old mails and accessing their attachments.



Chat Account Wizard

Chat network

Please choose the network of your chat account.

- Facebook Chat
- Google Talk
- IRC
- Twitter
- XMPP

» ***Thunderbird* also includes a small chat client, with support for Facebook messaging, Google Talk and IRC.**

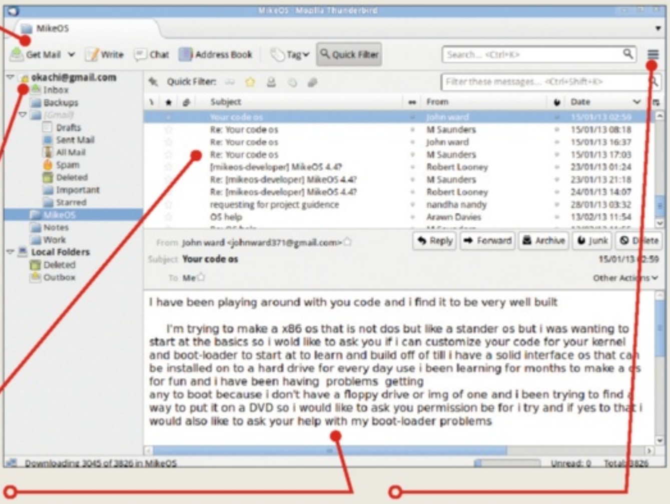
> **"Whole message threads can now be ignored or watched."**

Of course, you've probably heard of *Thunderbird*, as it's one of the most popular email clients in the world, and it's from the same team that makes *Firefox*. So we won't pore over every little feature here, but give you a quick tour of the new additions and interface. To run it, you will need to extract the **thunderbird-24.0.tar.bz2** file and run *Thunderbird* in the resulting directory. You don't need to install it system-wide, but if you prefer to, extract it into **/usr/lib/thunderbird/** and add a launcher or window manager.

When you first run *Thunderbird*, you're prompted to enter your email address and password, and the client will attempt to connect to the relevant email service. In many cases you can choose between IMAP or POP3 protocols – the former only downloads email headers and retrieves the contents on demand, whereas the latter downloads entire emails and attachments onto your hard drive.

Major new features in Thunderbird 24 include: whole message threads can now be ignored or watched; it's possible to zoom in and out in the Compose window (and not just change the font size); support for IDN email addresses has been added (ie international addresses using Unicode); and in IRC chat, user nicknames are highlighted when they are mentioned.

Exploring the Thunderbird interface



Subject	From	Date
Your code os	John ward	15/01/13 02:59
Re: Your code os	M Saunders	15/01/13 08:18
Re: Your code os	John ward	15/01/13 16:37
Re: Your code os	M Saunders	15/01/13 17:03
[mikeos-developer] MikeOS 4.4?	Robert Looney	23/01/13 01:24
Re: [mikeos-developer] MikeOS 4.4?	M Saunders	23/01/13 21:18
Re: [mikeos-developer] MikeOS 4.4?	Robert Looney	24/01/13 14:07
requesting for project guidence	nandha nandy	28/01/13 03:32
OS help	Arawn Davies	13/02/13 11:54

From John ward <johnward371@gmail.com>
Subject **Your code os** 15/01/13 02:59
To Me

I have been playing around with you code and i find it to be very well built

I'm trying to make a x86 os that is not dos but like a stander os but i was wanting to start at the basics so i wold like to ask you if i can customize your code for your kernel and boot-loader to start at to learn and build off of till i have a solid interface os that can be installed on to a hard drive for every day use i been learning for months to make a os for fun and i have been having problems getting any to boot because i don't have a floppy drive or img of one and i been trying to find a way to put it on a DVD so i would like to ask you permission be for i try and if yes to that i would also like to ask your help with my boot-loader problems

Downloading 3045 of 3826 in MikeOS   Unread: 0   Total: 3826

1. Tabs
The Default tab shows mail, but you can open others for different *Thunderbird* features, eg by clicking the Chat button.

2. Email accounts
Your email accounts and their respective folders are shown here. If you're accessing a Google Mail account, labels appear as folders.

3. Message list
Unread messages are marked by bold text. By default, the newest messages will appear at the bottom; this is changeable by clicking the Date column header.

4. Message view
Email contents are shown here. Unfortunately, *Thunderbird* can't change non-punctuated nonsense into anything readable.

5. Menu
Thunderbird's menu isn't displayed by default - access it via the three-line button here.

Dragon 32/64 emulator

XRoar

Version: 0.30.2 **Web:** http://bit.ly/18T0a10

› **Although the Dragon didn't have a huge game library, it had some notable titles, such as *Chuckie Egg*.**

We fondly remember the ZX Spectrum vs Commodore 64 flame wars of the 1980s. Well, they were largely face-to-face flame wars, as there wasn't much in the way of online communication as there is now. Occasionally, though, someone would join the debate and throw a curve ball into the proceedings. You'd be slapping down some C64 owner because he needed an external tape drive (whereas your lovely Spectrum +2A had one built in), and then someone would say "I own a Dragon 64, and it kicks both of your pathetic machines' bottoms."

The Dragon 32/64 (the numbers referring to the installed memory in k) were largely unknown 8-bit computers built in the early 1980s. Spec-wise they weren't massively interesting – a 6809 CPU clocking in at 0.89MHz and Microsoft BASIC in ROM – but they were made in Wales, hence the name. Even though the Dragon ceased production in 1984, the machine still has a small group of fans on the net, and an emulator in the form of *XRoar*.

To build it from source on Debian/Ubuntu-based distros, you'll need to grab the following dependencies:

```
sudo apt-get install build-essential
libsndfile1-dev libgtk2.0-dev
libgtkglext1-dev libasound2-dev
```

These are available under similar names in other distros. Then run **./configure**, **make** and **make install** (the last step as root). If you see a problem about a missing library in the last step, edit **config.mak** and change the LDFLAGS line so it looks like this:

```
LDFLAGS = -lm
```

> "The Dragon machine has an emulator in the form of XRoar."

Now run the build procedure again. Before you can use the emulator, you'll need to get a ROM image for a Dragon machine and place it in **/usr/local/share/xroar/**; due to potential copyright issues, we can't give exact locations on the web. Then run **xroar** to start it, optionally with **-default-machine** to specify the type of Dragon:

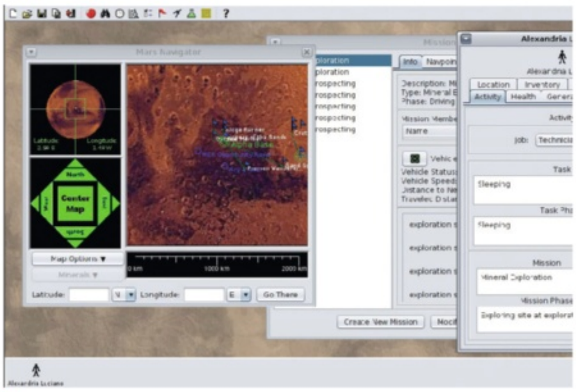
```
xroar -default-machine dragon32
```

From here you can hit Ctrl+Shift+L to load a tape (CAS, WAV or BAS) image or a disk (DMK or VDK) image, and auto-run it. A full list of keybindings can be found in the manual.

Mars settlement planner

Mars Simulation

Version: 3.05 **Web:** http://mars-sim.sourceforge.net



› **The Navigator panel enables you to see where your rovers are scuttling about on Mars.**

Whether we'll see a human base on Mars in the next few decades is anyone's guess, but plans are being made. The problem is, the job is immensely complicated. Many armchair astronauts assume that we can just do a replay of Curiosity and replace the rover with a couple of people, and Bob's your uncle – humans on the red planet. But the logistics are very different, the payload would have to be much heavier, and once you have people strutting around on Mars, what are they going to do? They need shelter, a way to grow food and so forth.

The *Mars Simulation Project* is a brilliantly detailed Java app which highlights the complexity of a human Mars mission. You can grab it as a ZIP file or tarball (run **java -jar mars-sim-main-3.05.jar** after extraction), and there's also a Deb for Ubuntu and Debian-based distros, which places a launcher under the Education menu in most desktops and window managers.

When you first start the sim, you're given a choice of two settlements to manage: the first has a population of 24 and is at a fairly advanced stage, and the second has only four people and needs more work to get going. Choose one and click Create New Simulation. A tutorial window will pop up and then you can start exploring the settlement.

The Simulation runs in accelerated time (10 minutes in the settlement pass in 1 second of real time); this can be changed under the Tools menu. This menu also provides access to various views of activity on the planet, so you can look at different science research missions, for instance, or check on the health of individuals. Much of the activity takes place without your direct involvement, but you can make high-impact changes, such as rescheduling supply missions and creating new tasks, such as constructing buildings or searching for minerals.

> "The app highlights the complexity of a human Mars mission."

Above all, it really hammers home how much planning is required if we ever hope to have a Mars base one day.

Lossless JPEG cropper

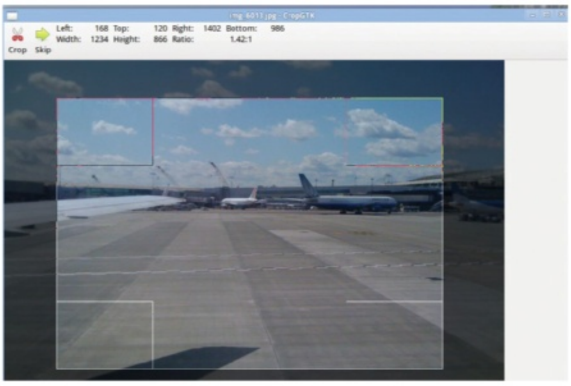
CropGUI

Version: 0.1.1 **Web:** http://bit.ly/1a5fLHW

Here's item #573 from our ever-growing list of pet peeves on the internet: people using PNG images for photos. OK, not everyone knows the differences between image formats, but far too often we see very large photos, which could be stored in a couple of MBs in JPEG format, presented via mammoth 10MB+ PNG images. Most **LXF** readers will know which format to use, but it bears repeating: JPEGs are ideally suited to photos because of lossy compression. They can scrub out bits that are barely recognisable to the eyes, in order to keep file sizes down. PNGs store every single pixel with lossless compression, so they're far better suited to diagrams, logos, pixel artwork and the like.

OK, rant over. If you have a bunch of JPEG images and want to crop them down to a specific region, you could open them up in *Gimp*, do the cropping and re-save them. However, each time you re-save, you have to specify the lossy compression level again, so the previously compressed picture is re-compressed. Over time, this leads to blurry and artefact-laden images – a bit like copying a cassette tape onto another cassette tape, and then that tape onto another tape, and so on. The audio becomes very muffly in the end.

CropGUI is a small Python tool that lets you crop areas of JPEG images without having to recompress them, avoiding this problem. There are two versions included in the tarball: **cropgui.py** is the older version which uses *Tk* for the interface, whereas **cropgtk.py** has a more polished *GTK*-based front-end. When you start the program, you're prompted to locate a file, choose one, and then drag the corners of the on-screen rectangle to the specific area that you want to save.


img 6011.jpg - CropGTK

Left: 168 Top: 120 Right: 1402 Bottom: 986
Width: 1234 Height: 866 Ratio: 1.42:1

› Amazingly, cropping a screenshot showing a cropping operation didn't destroy the space-time continuum.

And here's where you might get lost. There's no Save button in the interface, and when you click Crop, you're immediately presented with the file chooser (to open a new file). Worry not, though, because the previously cropped file will be saved with a new filename, so if you opened **image_4734.jpg** then the smaller version will be saved as **image_4734-crop.jpg** automatically. Once you have this sussed out, the program is simple and very fast to use.

> "Crop areas of JPEG images without having to recompress them."

Photo manager

DigiKam

Version: 3.4 **Web:** www.digikam.org

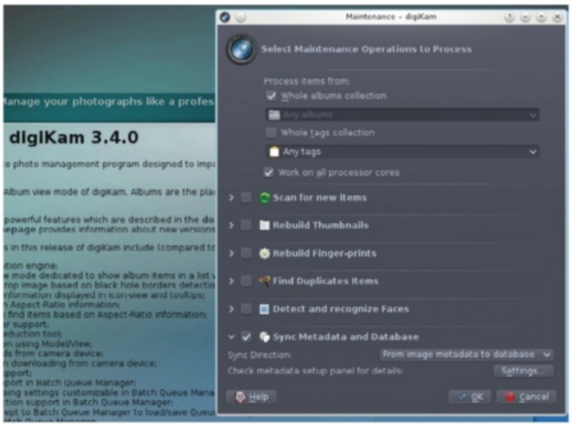
The whole KDE vs Gnome argument ultimately boils down to configurability vs simplicity. KDE fans accuse Gnome developers of overly dumbing down the interface and removing important options, while Gnomers regard KDE as a nightmarish labyrinth of settings and buttons. And this debate extends beyond the desktop environments to many of their major applications, such as *DigiKam* (KDE) vs *Shotwell* (Gnome). To quote Adam Oxford (*Roundup*, **LXF168**): "Where *Shotwell*'s editing tools are too basic for most, *DigiKam* can be overwhelming."

For all its complexity, *DigiKam* is a hugely popular photo manager. Here we'll focus on installing it and exploring the new features in 3.4. If no packages are available in your distro's repositories, you'll have to compile it from source, which isn't so painful providing you have the right dependencies (*libqt*, *kdelibs*, *libkipi*, *libkdcraw*, *libkexiv2*, *libgphoto2*, *liblcms*, *libjasper*, *libtiff* and *libpng*). See the **Readme** file for the versions required.

With everything in place, build and install *DigiKam* with these commands:

```
cmake .
make
sudo make install
```

You can add **-DCMAKE_INSTALL_PREFIX** to the **cmake** command to specify a different location for installation (**/usr/local/** is the default).


Maintenance - digiKam

Select Maintenance Operations to Process

Process items from:
☑ Whole albums collection
Any albums
☐ Whole tags collection
Any tags
☑ Work on all processor cores
☐ Scan for new items
☐ Rebuild Thumbnails
☐ Rebuild Finger-prints
☐ Find Duplicates Items
☐ Detect and recognize Faces
☑ Sync Metadata and Database
Sync Direction: From image metadata to database
Check metadata setup panel for details: Settings...

› In version 3.4 it's now possible to select specific albums and tags in the Maintenance Tool.

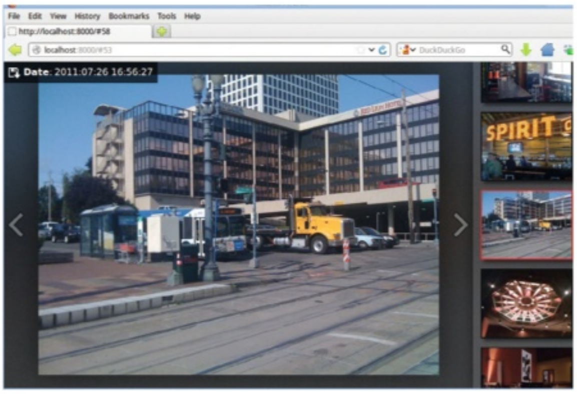
The biggest new feature in version 3.4 is better support for files generated by Nikon's *Capture NX* photo editing program: *DigiKam* can now read their Color Label information. In the Maintenance Tool, there are new options for selecting albums and tags to process, while performance has been boosted thanks to multi-threading and multi-core CPU support in the Metadata Synchroniser, Thumbs Generator and Fingerprint Generator tools. Smaller changes include different defaults in the noise reduction tool, fixes to the Face Recognition feature, and small bugfixes and code cleanups.

> "Supports files generated by Nikon's Capture NX photo editing program."

Photo gallery generator

Fgallery

Version 1.0 **Web:** http://bit.ly/18pukc1



› **As well as using the arrow icons, you can navigate between images using the cursor keys.**

Whatever happened to sensible user interface design? Everything seems to have gone bonkers in the last few years. Gnome ditched decades of well-established concepts in favour of the most dumbed-down thing imaginable, while Microsoft decided that everyone really wants a tablet interface, even on a 32-inch monitor. Even Apple's latest iOS looks like something you'd see in Playmobil-land.

So discovering *Fgallery* has made us very happy pandas indeed. It's a Perl program that generates web-based photo galleries from your image collections, and the interface for these galleries has the perfect balance of simplicity, functionality and style. It focuses almost entirely on navigation and displaying images, with just a smattering of information displayed.

If you're running Debian/Ubuntu, get the required dependencies like so:

```
sudo apt-get install imagemagick exiftran zip libjson-perl libjson-xs-perl libtimedate-perl
```

These packages should be available in other major distros as well. Once you have everything in place, extract **fgallery-1.0.zip** and then copy the contents of the View directory inside it into a destination folder for your gallery (eg **web-upload**). Then run:

```
./fgallery /path/to/photos /path/to/web-upload
```

If you now go into **/path/to/web-upload**, you'll see **index.html** – but don't try to open it in your web browser. Because *Fgallery* makes use of Ajax, which usually isn't supported for locally stored files in browsers, you'll need to make **/path/to/web-upload** available via a web server somewhere. A quick way to do this is to jump into the directory and enter:

```
python -m SimpleHTTPServer 8000
```

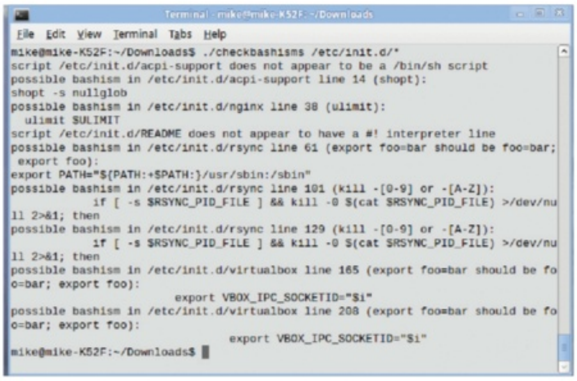
Visit **http://localhost:8000** to view the gallery in your browser. *Fgallery* automatically rescales and compresses images to make them appear faster over the web, and it also strips out EXIF data. The interface is slick and simple: thumbnails on the right, full pictures on the left, and the date information displayed above. You'll also see a floppy disk icon – this is a link to a ZIP file of the original, unrescaled images.

> "Fgallery has the perfect balance of simplicity, functionality and style."

Script portability checker

Checkbashisms

Version: 2.0.0.2 **Web:** http://bit.ly/19W0hLm


Terminal - mike@mike-K52F: ~/Downloads
File Edit View Terminal Tabs Help

```
mike@mike-K52F:~/Downloads$ ./checkbashisms /etc/init.d/*
script /etc/init.d/acpi-support does not appear to be a /bin/sh script
possible bashism in /etc/init.d/acpi-support line 14 (shopt):
shopt -s nullglob
possible bashism in /etc/init.d/nginx line 38 (ulimit):
  ulimit $ULIMIT
script /etc/init.d/README does not appear to have a #! interpreter line
possible bashism in /etc/init.d/rsync line 61 (export foo=bar should be foo=bar;
 export foo):
export PATH="${PATH:+$PATH:}/usr/sbin:/sbin"
possible bashism in /etc/init.d/rsync line 101 (kill -[0-9] or -[A-Z]):
            if [ -s $RSYNC_PID_FILE ] && kill -0 $(cat $RSYNC_PID_FILE) >/dev/nu
ll 2>&1; then
possible bashism in /etc/init.d/rsync line 129 (kill -[0-9] or -[A-Z]):
            if [ -s $RSYNC_PID_FILE ] && kill -0 $(cat $RSYNC_PID_FILE) >/dev/nu
ll 2>&1; then
possible bashism in /etc/init.d/virtualbox line 165 (export foo=bar should be fo
o=bar; export foo):
                        export VBOX_IPC_SOCKETID="$i"
possible bashism in /etc/init.d/virtualbox line 208 (export foo=bar should be fo
o=bar; export foo):
                                export VBOX_IPC_SOCKETID="$i"
mike@mike-K52F:~/Downloads$ 
```

› **We won't give Ubuntu a hard time for having *Bash*-specific scripts, as it makes sense here, but this example shows what problems can crop up.**

Like us, you probably spend most of your time in Linux, but enjoy the occasional foray into other open source or Unix-like operating systems – FreeBSD, NetBSD, maybe even a bit of Solaris or AIX. And there's a recurring theme when you read the forums and newsgroups for these OSes: users complain that too much open source software is Linux-specific. Instead of coding to OS-neutral standards, such as POSIX, many developers assume that everyone is running Linux, and add lots of Linux-specific bits to their code.

A good example of this is with shell scripts. *Bash* is pretty much the de-facto standard on Linux, but you can't guarantee this across distros, and on other Unix-like OSes the default shell (**/bin/sh**) probably won't be *Bash* (even if *Bash* is available or installed elsewhere). So if you're writing a script and you want it to run everywhere, you need to avoid adding *Bash*-specific code. This is easier said than done, but *Checkbashisms* can help. It's a Perl script which roots around inside *Bash* scripts to find potential compatibility problems with a vanilla **/bin/sh** *Bourne* shell. To get it working, download it and make it executable (**chmod +x checkbashisms**). You can then run it in place with one or more shell scripts as a parameter – for instance, to see how *Bash*-specific your distro boot scripts are, enter:

```
./checkbashisms /etc/init.d/*
```

Now *Checkbashisms* will show potentially problematic lines, and often provide alternatives, such as:

```
export foo=bar should be foo=bar; export foo
```

In this case it's a syntactical problem, and you'll also see warnings about built-in commands that may not be provided by a standard **/bin/sh** implementation. You shouldn't use **shopt** and **alias**, for instance.

A lot of the time, remedying these issues will involve making your code longer and more complicated, and you'll feel like you're using a more primitive shell, but that's the price you pay for better portability.

> "Checkbashisms will show potentially problematic lines."

HotGames Entertainment apps

Top-down racer

Dust Racing 2D

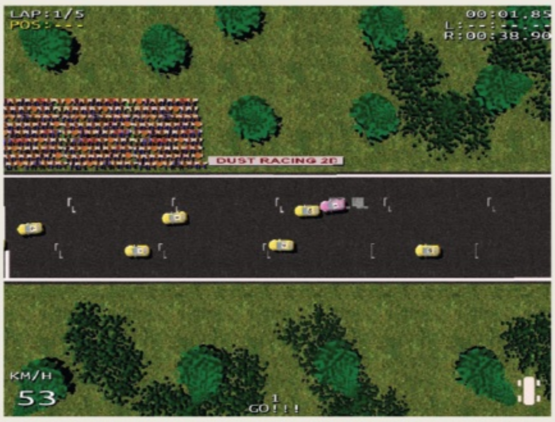
Version: 1.4.3 **Web:** http://bit.ly/17xKRqT

If you're an older **LXF** reader then you may remember *Super Sprint*. This innovative arcade racer featured steering wheels on the cabinet, and supported three players simultaneously. It was a simple game: you had a static, top-down view of a race track, and your goal was to zip around it, overtake the competition and win in three laps.

Dust Racing 2D owes a lot to *Super Sprint*, but it's considerably more tarted-up, with a scrolling view of the track and detailed pixel artwork. Helpfully, you don't have to compile it from source: along with packages for Debian/Ubuntu-based distros, there are pre-built binary tarballs. Grab **dustrac-1.4.3-linux-i686-qt4.tar.gz** (or the x86_64 alternative if you're running a 64-bit distro), extract it and run **./dustrac-game** in the resulting directory.

Hit Enter on the Play menu to get started. Initially, only a single track can be accessed, but more will become playable as you win races. At the start of each race your pink car is at the back of the grid, behind 11 CPU-controlled opponents – and they mean business. Use the Up arrow key to accelerate, Down to break, and Left and Right to steer. You might find it nightmarishly difficult at the beginning, as the car's handling is very slippy, but after a while you'll achieve the appropriate level of twitch response to make it round corners without skidding off the track.

To move onto the next track, finish in the top six. There are only six courses included in the game (a couple loosely modelled on real Formula 1 tracks) but you can add your own via the included level editor. Don't go fiddling with your audio settings when you hear nothing in the game, though – for once, it's not PulseAudio falling to pieces again. No, *Dust Racing 2D* is simply eerily silent.



› **The background scenery is largely static, save for some gently swaying trees.**

> **"Dust Racing 2D owes a lot to Super Sprint, but it's more tarted-up."**

Maze-exploring romp

Chrzazscz

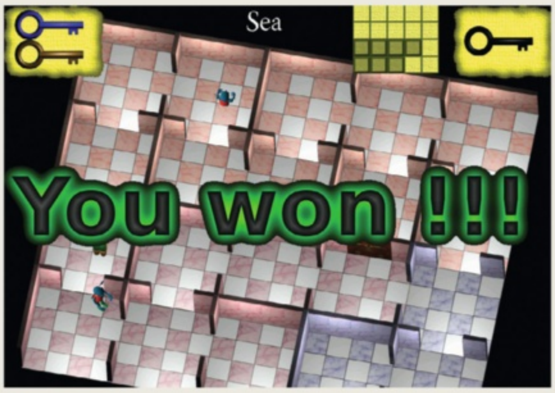
Version: 0.1.0 **Web:** http://bit.ly/H7Okpb

If you're Polish, congratulations; you can probably pronounce this with ease. Please do us a favour: go to **www.vocaroo.com**, record yourself saying the name and share the result on our forums. But anyway, *Chrzazscz* is an early-in-development maze exploration game built with the SDL libraries, and to compile it from source you'll also need *SDL-mixer*, *SDL-image* and *SDL-ttf*.

After the tortured-guitar music of the intro screen, *Chrzazscz* is, like *Dust Racing 2D*, completely silent. Hit Enter to select a level to play – five are included, and you can access them all right from the start. You're then prompted to choose which character you'd like to play as (a fighter, thief or wizard) and then you're dropped into the maze. At first you'll only see a single room, but as you walk through doorways, your visible part of the map will open up.

There's no battling or magic spell casting to talk of here; it's all about exploring. Hit F1 to show your list of objectives for the level, which always involve visiting certain areas of the maze. Some of these areas are blocked by locked doors, and to get the corresponding colour-coded keys, you have to speak to blue orcs running around. However, you can only hold two keys at any one time, and some orcs will only swap keys with specific colours, so you have to plan your movements in advance. It doesn't help that the orcs can disappear into unvisited (and therefore invisible) rooms as well...

Chrzazscz is still very early in development and many elements of the game are absent right now, but it shows a lot of promise and is already playable. With a few more levels, more work on the presentation side and greater variety between the player types (right now the differences appear to be entirely cosmetic), this could become a classic one day.



› **Beating a level is satisfying enough, but the three exclamation marks add a whole other level of euphoria.**

> **"Chrzazscz shows a lot of promise and is already playable."**

Quick directory switcher

bd

Version: N/A **Web:** https://github.com/vigneshwaranr/bd

While we like discovering big and shiny new apps for HotPicks, we also love stumbling across nifty little time-savers. A perfect example is *bd*: it's a small shell script that doesn't do anything spectacular, but over the course of weeks and months, it could save you a lot of time if you're a frequent user of the command line.

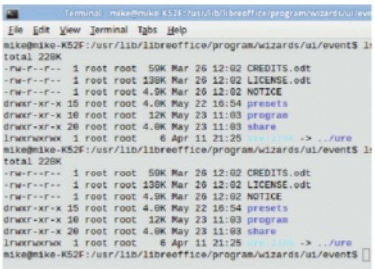
Here's how it works. Imagine you're in a deeply nested directory, such as **/usr/lib/libreoffice/program/wizards/ui/event**, and you want to jump back up a few levels. To switch to **/usr/lib/libreoffice**, you'd have to enter this:

```
cd ../../../..
```

That's not very elegant. And you have to keep calculating which directory you're jumping into, for each pair of dots you enter. Wouldn't it be far easier if you could just type the standalone name of the higher-level directory? Well, that's what *bd* lets you do:

```
bd libreoffice
```

Here, *bd* jumps you back into **/usr/lib/libreoffice/** without you having to specify the full path or join lots of dots together. If there are no name clashes between directories, you can even avoid typing the full thing, so **bd libre** has exactly the same effect. If you spend a lot of time working in deeply nested directories (which is usually the case for developers working on large codebases), this should be an essential weapon in your CLI armoury.


Terminal - mike@mike-K52F: /usr/lib/libreoffice/program/wizards/ui/eve
File Edit View Terminal Tabs Help

```
mike@mike-K52F:/usr/lib/libreoffice/program/wizards/ui/event$ l
total 228K
-rw-r--r--  1 root root  59K Mar 26 12:02 CREDITS.odt
-rw-r--r--  1 root root 138K Mar 26 12:02 LICENSE.odt
-rw-r--r--  1 root root 4.9K Mar 26 12:02 NOTICE
drwxr-xr-x 15 root root 4.0K May 22 16:54 presets
drwxr-xr-x 10 root root  12K May 23 11:03 program
drwxr-xr-x 20 root root 4.0K May 23 11:03 share
lrwxrwxrwx  1 root root    6 Apr 11 21:25 ure-link -> ../ure
mike@mike-K52F:/usr/lib/libreoffice/program/wizards/ui/event$ l
total 228K
-rw-r--r--  1 root root  59K Mar 26 12:02 CREDITS.odt
-rw-r--r--  1 root root 138K Mar 26 12:02 LICENSE.odt
-rw-r--r--  1 root root 4.9K Mar 26 12:02 NOTICE
drwxr-xr-x 15 root root 4.0K May 22 16:54 presets
drwxr-xr-x 10 root root  12K May 23 11:03 program
drwxr-xr-x 20 root root 4.0K May 23 11:03 share
lrwxrwxrwx  1 root root    6 Apr 11 21:25 ure-link -> ../ure
mike@mike-K52F:/usr/lib/libreoffice/program/wizards/ui/event$
```

› Use backticks to list files contained in higher-level directories: eg ls -l `bd libre`.

Internet suite

SeaMonkey

Version: 2.21 **Web:** www.seamonkey-project.org

It's not dead! *SeaMonkey* is an incredibly low-profile project these days, despite it having a fascinating history. For those who haven't come across it before: *SeaMonkey* is the follow-up to the Mozilla suite, which in turn was the successor to Netscape, the browser that dominated home computers in the late 1990s (before Microsoft shoved Internet Explorer down everyone's throats).

Today all the buzz surrounds two standalone programs forked from the Mozilla codebase, *Firefox* and *Thunderbird*, but development has been quietly progressing on *SeaMonkey*. The project combines multiple internet tools into a single app, which some may describe as bloated and a violation of the Unix philosophy, while others find it useful to have everything in one place: SeaMonkey is comprised of a web browser, email client, IRC chat program and HTML editor. The GUI layout isn't drastically different from the days of Mozilla and Netscape, so if you long for a browser that doesn't try to dumb everything down and remove buttons in the name of 'simplicity', give it a go.

It's really easy to install as well: just grab **seamonkey-2.21.tar.bz2**, extract it, and run **seamonkey** in the resulting directory. There's no need to install it system-wide, and the program will pick up your *Thunderbird* settings if applicable. On first launch, it will also offer to become the default browser. LXF

› ***SeaMonkey*** **displaying a seven year-old screenshot of the Mozilla suite – note how the GUI layouts are almost identical.**

Also released

New and updated software that also deserves a look...

Beets 1.3.0
Improve your music collection with better metadata.
beets.radbox.org

deheader 0.8
Avoid duplicated header file inclusion in C/C++ source code.
www.catb.org/~esr/deheader/

BirdFont 0.30
Font editor which can create TTF, EOT, and SVG fonts.
http://birdfont.org

› See LXF175's HotPicks for a closer look at this app.

IceWM 1.3.7
Classic window manager, not as dead as some people expected.
www.icewm.org

Gnome Chemistry Utils 0.14.2
Suite with a chemical editor, calculator and 3D molecule viewer.
http://gchemutils.nongnu.org

Emacs 24.3
Much-loved/hated GNU editor, now with a packaging system.
www.gnu.org/software/emacs/

F-IRC 1.10
Console-based IRC client that aims to be easy to use and navigate.
www.vanheusden.com/fi/

Rescue! Max
An action-adventure space game written in Java.
http://rescue.sourceforge.net

› All looks very -esque, doesn't it?

Back issues » Missed one?

Issue 177
December 2013

Product code:
LXFDB0177

In the magazine

Uncover the next generation of open source mobile OSes in Ben Everard's last feature for **LXF**. Go inside Project Sputnik: Dell's Ubuntu laptop and discover the programs teaching kids to code.

LXFDVD highlights

LXLE, OS/4 OpenLinux and Grml in 32- and 64-bit flavours.

Issue 176
November 2013

Product code:
LXFDB0176

In the magazine

Not satisfied with the plethora of Linux distros available? We show you how to build your own. Discover why you'll want an Ubuntu phone and reveal the Raspberry Pi extras that you'll want to tinker with at home.

LXFDVD highlights

Rosa Desktop Fresh R1 64-bit, Sonar Linux and 4MLinux 7.

Issue 175
October 2013

Product code:
LXFDB0175

In the magazine

Level up your skills with our guide to the tools of tomorrow. Peek into the world of Arch Linux, and get the inside track on how open source is being put to good use at Jolla, the successor to Meego, the mobile OS.

LXFDVD highlights

Fedora 19, Arch Linux, Knoppix, Tails Linux and Overclockix.

Issue 174
September 2013

Product code:
LXFDB0174

In the magazine

Save time and work smarter with the right desktop for you. Explore the Raspberry Jam, marvel at the insight of Mozilla's top security guy, and feed your brain with tutorials on Puppet, Kotlin, Grub 2 and more.

LXFDVD highlights

Gnome Tweak Tool for Gnome 3, Mint 15, Linux Bodhi, Siduction.

Debian 7 | Mageia 3 | Sabayon
LINUX FORMAT
#1 for Free Software
Escape Google
And Facebook » Twitter » Flickr and more!

Issue 173
August 2013

Product code:
LXFDB0173

In the magazine

Learn how to liberate your data from Google. Discover Mir, the controversial display graphics project and delve inside the EFF, Ruby on Rails, Raspbian, MariaDB, bracket-loving Lisp and more.

LXFDVD highlights

Debian 7, Mageia, Sabayon and a blast from the past: issue **LXF64**.

Issue 172
July 2013

Product code:
LXFDB0172

In the magazine

We ask 'Dude, what's happened to Ubuntu?' Plus the ethos behind distinctly Gaulish distro, Trisquel, Firefox OS phones, Scratch, Kerbal Space Program and finding dead people with Gramps genealogy app.

LXFDVD highlights

Our Privacy Enhanced remix of Ubuntu, Kubuntu, Manjaro.

To order, visit www.myfavouritemagazines.co.uk

Select Computer from the Browse Magazines list and then select Linux Format.

Or call the back issues hotline on 0844 848 2852 or +44 1604 251045 for overseas orders

Quote the issue code shown above and have your credit or debit card details ready

Future

GET OUR DIGITAL EDITION!

SUBSCRIBE TODAY AND GET 2 FREE ISSUES

Available on your device now

Available on the App Store | amazon apps Available at amazon | GET THE MAGAZINE ON Google play | nook | zinio™

*Free Trial not available on Zinio.

LINUX FORMAT Tutorials

Turn on, log in and let our experts help make you a smarter Linux user

ANDREW GREGORY is overcome with emotion at the thought of leaving LXF Towers.

This is the end

The time has come to depart. Ben, Graham and myself have all taken the decision to leave *Linux Format* and Future Publishing.

Since my 2005 introduction to Mandrake (the forerunner to Mandriva/Mageia/Rosa), Linux has changed beyond measure. Installing codecs to watch DVDs used to be a subject for tutorials in **LXF**; now it's a matter of ticking a box at install time. Ubuntu had produced only one release, and was little more than a more streamlined version of Debian. Games were limited to *SuperTuxKart*, media keys worked at random, and the perception of Linux as a geek hobby for weirdos was hard to shake off. All of this has changed.

All change

One thing that hasn't changed in all that time is the strength behind Linux: its users. The podcasts, forums, wikis and events such as OggCamp generated by free software advocates show how strongly people feel about it. That, in a nutshell, is why Linux has been able to adapt and thrive, and why it will continue to do so in the coming years.

To everyone who has ever emailed us with a suggestion, read the magazine, listened to the podcast, asked a question on the forums or commented on TuxRadar.com, thank you. We're privileged to have been part of such a fantastic community.
@AndrewGregory83

This month learn how to...

Arrange photos...........68

Matthew Hanson gets to grips with *Shotwell*, the number one image collection manager for Linux, to best show off his 'arty' pictures.

Fortify your servers...70

Jonathan Roberts has a grown-up job now, and has shared with us the secrects of managing high-availability servers.

Manage disk space....74

It's easy to accumulate cruft on your hard disk, so you don't need your operating system to clog it up for you. **Neil Bothwick** is the disk janitor.

Craft a desktop............ 76

Take a window manager, a panel, a few choice extras and some guidance from **Mike Saunders** and you have your own custom desktop.

Deploy Drupal 80

Harness the supreme power of the Drupal content management system the ~~lazy~~ efficient way with **Shashank 'efficient' Sharma**.

Code in tutorials

Source code is presented in beige boxes. When lines of code are too long for our columns, the remaining text appears on the next line in a solid box:

```
procedure TfrmTextEditor.
mniWordWrapClick
```

Otherwise, there is a gap, like this:

```
begin
  mniWordWrap.Checked := false
```

CODING ACADEMY

Whether you're a beginner or a guru, learn new skills in our programming section

In this issue...

Python

Get in at ground level with this primer to the very basics of Python programming **p84**

Fortran

Computer science's version of a stone hand-axe is still nice and sharp. And very old... **p90**

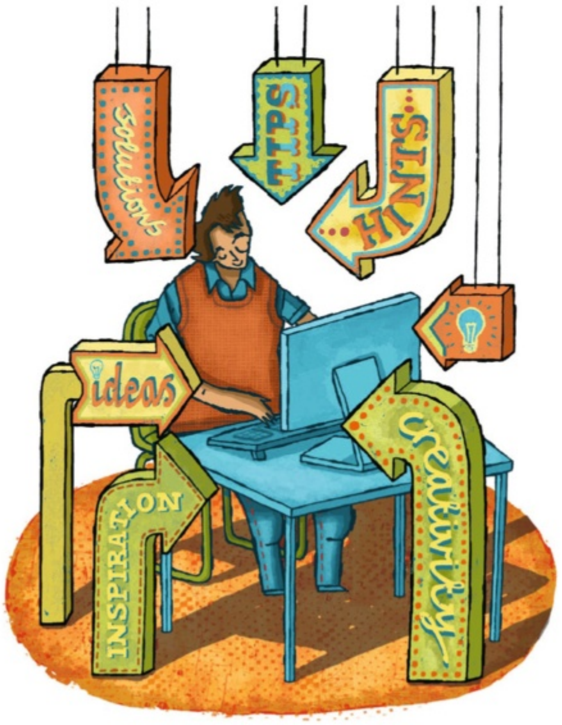
Shotwell A quick guide to importing, organising and editing your photos

Shotwell: photo management

Matt Hanson takes a tour of the user-friendly photo organiser to see how easy it is to import and edit his holiday snaps.

Our expert

Matt Hanson has been messing around with OSes since the Amiga Workbench 2.0. After dabbling in DOS & Windows, he's now turned his attention to Linux.

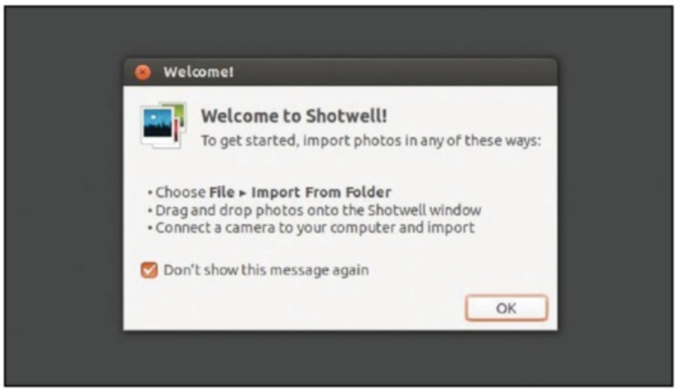


When *Shotwell* first appeared in 2009 it quickly proved itself to be an excellent open source alternative to propriety photo management software, such as Windows Live Galley and iPhoto. While copying your photos to your PC from your camera or phone is a simple enough process of dragging and dropping files, Shotwell has a nice graphical user interface that enables you to quickly import and organise your photos. You can group them together by date or through other metadata, such as location information.

As well as easily importing your photos from practically any device you plug in, you can quickly add metadata by 'tagging' the files. For example you could tag photos from a recent trip to sunny Oxford as 'holiday, Oxford'. This creates two tags – 'holiday' which would group the tagged photos with any other holiday snaps and 'Oxford' which would give you a more specific search criteria.

Shotwell also includes some basic – yet versatile – editing tools for smartening up photos, as well as options for quickly publishing and sharing photos and videos to online services, such as Flickr and YouTube. *Shotwell*, then, is a great app for taking what can be a tedious and time consuming process and turn it into something quick and straightforward. LXF

Organise your images in Shotwell


Welcome!

Welcome to Shotwell!

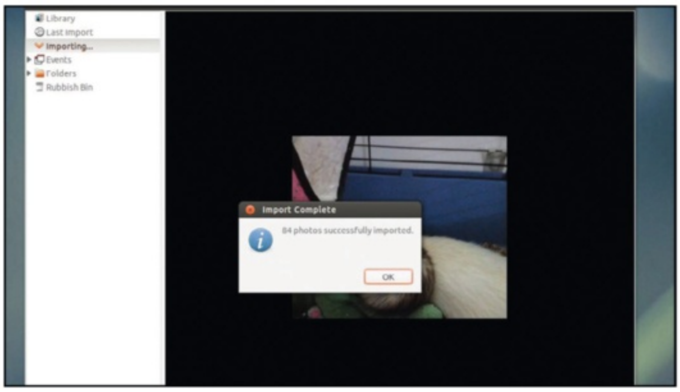
To get started, import photos in any of these ways:

- Choose **File ▸ Import From Folder**
- Drag and drop photos onto the Shotwell window
- Connect a camera to your computer and import

☑ Don't show this message again

1 Import photos from your PC

Shotwell replaced *F-Spot* as the standard image tool for many distros that use Gnome a while ago, so there's a good chance you already have it pre-installed. When loading it up for the first time you're prompted to import your files from the **home/user/Pictures** folder. If your photos are stored elsewhere you can browse to another folder to include them.

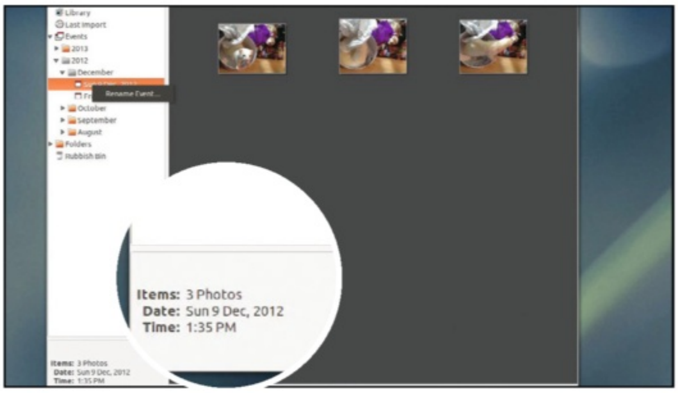


Import Complete

84 photos successfully imported.

2 Import files from a camera or phone

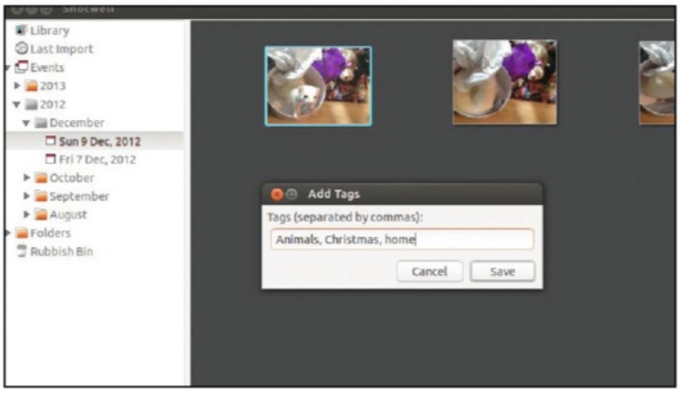
Some digital cameras, and almost all smartphones, come with their own bloated image editing and/or management software that insists on spreading itself through your computer, taking over as the default software for many tasks and for opening many file types. Thankfully, *Shotwell* can take the images and import them straight from your preferred device.



Items: 3 Photos
Date: Sun 9 Dec, 2012
Time: 1:35 PM

3 Shotwell Events

When the photos are imported into *Shotwell*, they are organised based on Events, which, by default, is the date and time the photo was taken (or when it was imported if there's no date and time information). If you right-click on an event you can give it a more memorable name.



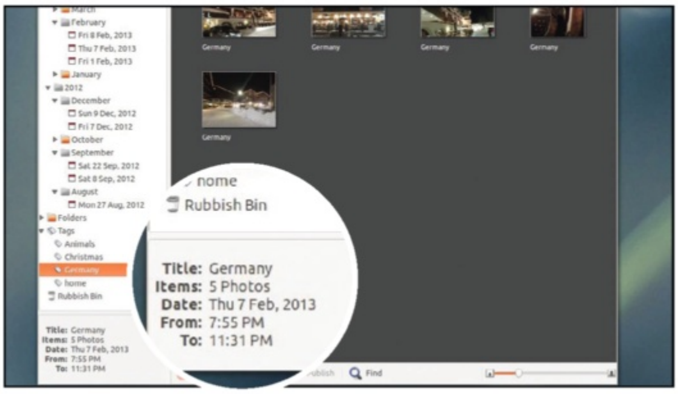
Add Tags

Tags (separated by commas):

Animals, Christmas, home

4 Adding metadata

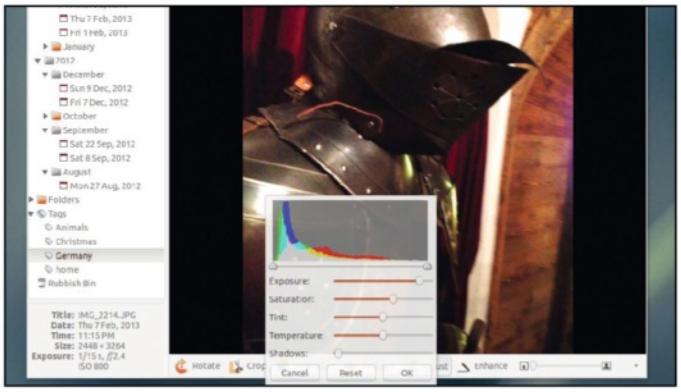
One of the easiest ways of adding additional information to your photos for easier organisation is to right-click on a photo and select Add Tags and then enter in the tags separated by commas. The more tags you add, the easier it will be to find the photos later. You can batch select photos to tag by selecting them before right-clicking.



Title: Germany
Items: 5 Photos
Date: Thu 7 Feb, 2013
From: 7:55 PM
To: 11:31 PM

5 Searching by metadata

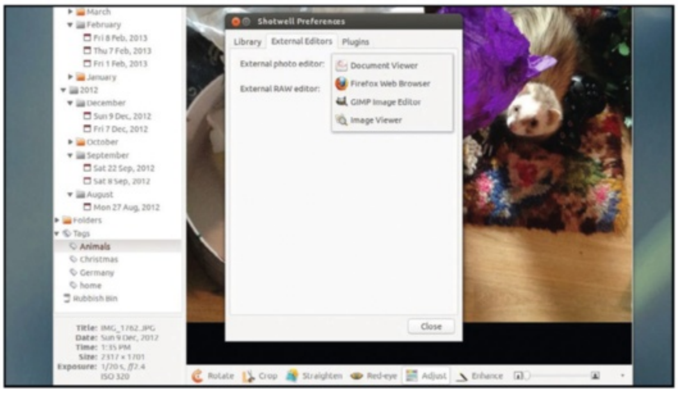
Along the left-hand side of the window you will see the various events listed and dates they were made. If you want to change information about when a particular photo was taken, right-click the photo and select Adjust time and date. Any tags that you have added will appear below the dates. Clicking on any tag will display the photos that are tagged with it.



Title: IMG_2214.JPG
Date: Thu 7 Feb, 2013
Time: 11:15 PM
Size: 2448 × 3264
Exposure: 1/15 s, f/2.4
ISO 800

6 Editing photos

Double-click on a photo and you'll be able to quickly edit and tweak the photo with a number of useful tools. Rotation and Cropping tools are handy for correcting badly composed shots, while the red eye reduction and exposure, saturation, tint and temperature tools give you more options for improving the quality of your photos. There's also a one-click fix button that automatically adjusts your images.



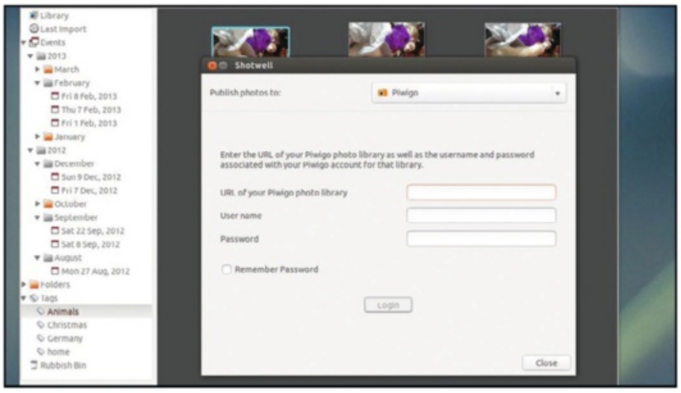
Shotwell Preferences

External photo editor:

External RAW editor:

7 External editing

The editing tools in *Shotwell* are perfectly acceptable for some quick tweaking, but if you're a serious photographer you'll want to use some more powerful photo editing tools. Click 'Edit' in the Toolbar, then select 'Preferences' and click on the 'External Editors' to choose other software, whichever you prefer to use such as GIMP, to edit your photos and RAW images.



Shotwell

Publish photos to: Piwigo

Enter the URL of your Piwigo photo library as well as the username and password associated with your Piwigo account for that library.

URL of your Piwigo photo library

User name

Password

□ Remember Password

8 Export and share

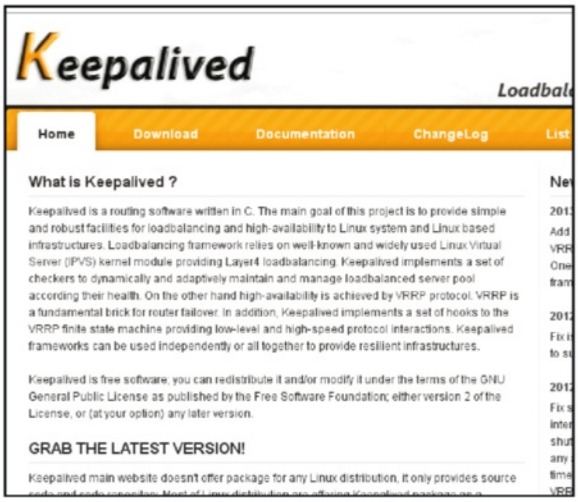
To publish and share a photo online, select a photo then click on the Publish button. A list of various online services you can upload to are in the top right-hand side of the window. Select the service you want, then click the login button to log in to your chosen service. Depending on the service you chose you'll see options for creating a new album, choosing privacy settings and the upload size.

High availability

Jonathan Roberts explains how to avoid disaster in the workplace when your servers crash, using processes such as load balancing.

Our expert

Jonathan Roberts has run away from *Linux Format* Towers to seek his fortune as a sysadmin.



Keepalived

What is Keepalived ?

Keepalived is a routing software written in C. The main goal of this project is to provide simple and robust facilities for loadbalancing and high-availability to Linux system and Linux based infrastructures. Loadbalancing framework relies on well-known and widely used Linux Virtual Server (IPVS) kernel module providing Layer4 loadbalancing. Keepalived implements a set of checkers to dynamically and adaptively maintain and manage loadbalanced server pool according their health. On the other hand high-availability is achieved by VRRP protocol. VRRP is a fundamental brick for router failover. In addition, Keepalived implements a set of hooks to the VRRP finite state machine providing low-level and high-speed protocol interactions. Keepalived frameworks can be used independently or all together to provide resilient infrastructures.

Keepalived is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

GRAB THE LATEST VERSION!

Keepalived main website doesn't offer package for any Linux distribution, it only provides source

› *Keepalived* **provides a simple high availability solution for web servers and load balancers, side-stepping the chicken and egg question.**

If you're in charge of running servers in a 'production' setting, it's likely that the business, organisation or person you're doing this for, has become dependent on the services you supply. They expect them to be available 24 hours a day, seven days a week, 365 days a year.

You live in the real world, however, and you know that computers are fickle things. Bugs exist in all software, and they always reveal themselves at the most inopportune moment; bringing down your mail server as the CEO tries to send a vital email, corrupting your database just after landing your biggest client, or your whole network crashing at 9am as all your employees sit down to start the working day.

We will introduce you to some simple techniques that will allow you to meet your dependants' expectations of continuous availability, while recognising that all hardware and software fails at some point.

No matter what the technique, the basic idea is the same: you have multiple servers for each kind of service you're running – five web servers, a pair of read-only database replicas (you could have a master-master pair for writes, too, but you have to be awfully careful when doing this, otherwise you could find yourself with inconsistent data), etc. If one server in the cluster fails, it doesn't matter; there's still one, or even several others running, which users can continue to access. If one of your servers does fail, you have lost some redundancy, but your services are still available and you can repair the broken component in a less frantic environment.

This idea of redundancy is at the core of all high-vailability (HA) setups – the question that you're left with then, is how do you implement such a redundant system?

The first technique we're going to introduce you to is called load balancing, and it's easy to configure for a wide range of circumstances. To demonstrate, let's take one of the easiest HA situations – a web server.

Two web servers are better than one

When you configure your web server, don't stop with just one; instead, set up a second and ensure both have exactly the same content and configuration. If this sounds tricky, you can read our Puppet tutorial in **LXF174**. You'll know exactly how you can configure identical servers easily and reliably. While configuring them, no doubt you'll have given each one a static IP address, so you know exactly where you need to go to get their content.

Ordinarily, you'd also enter these IP addresses in to your DNS system, giving each server a unique name, such as www1 and www2. This is great, but what address do you give out to your users or customers: **www1.mycompany.com** or **www2.mycompany.com**? If you care about HA, you'd have to give them both addresses – if the first fails, your users will have to take it upon themselves to remember the second server, then take the decision to use it.

This is not ideal – the less your users need to think about how to interact with your servers, the better it is for all concerned. There are two ways you can avoid this.

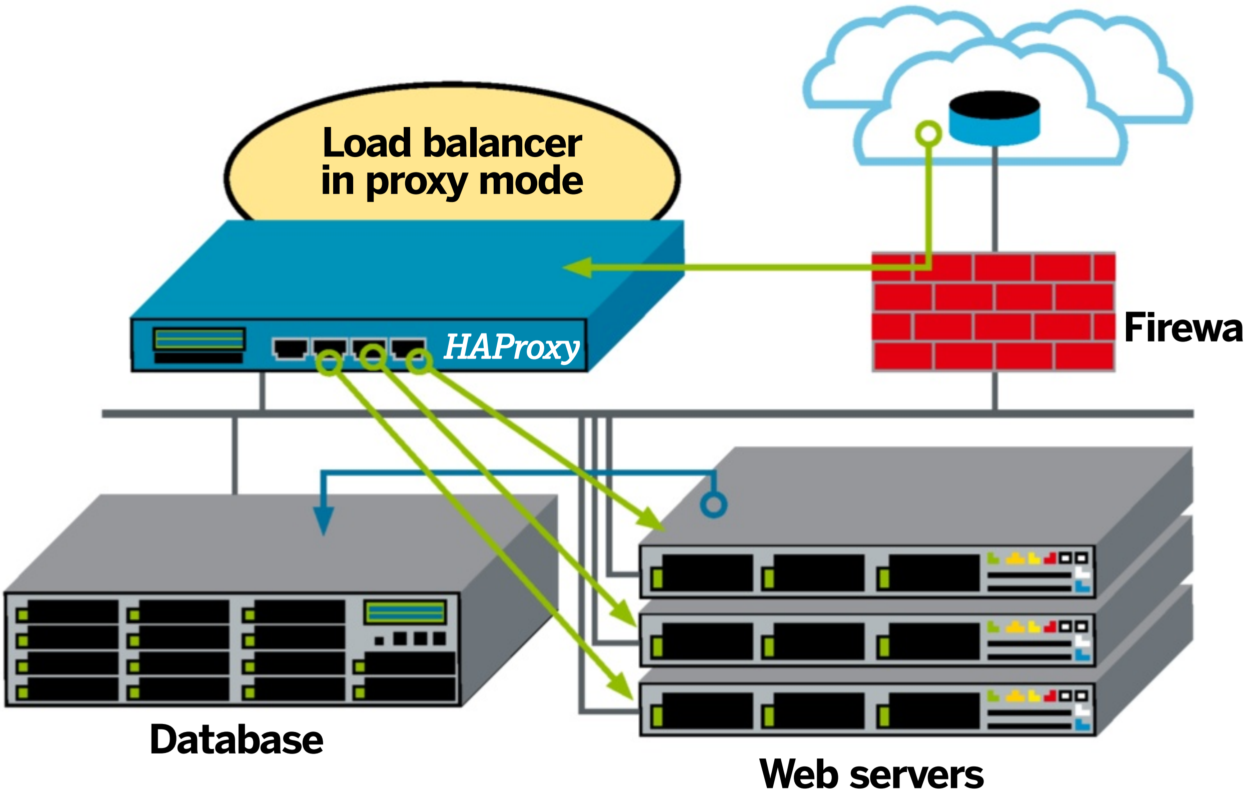
The first is called round-robin DNS. Rather than giving the

SPOFs

A contender for the most ridiculous acronym award, SPOFs are a key concept in HA setups. It stands for 'single point of failure', and if you care about up time, you need to develop a sixth-sense for these and erradicate them wherever they appear.

They're self-explanatory, really: a single point of failure is any part of an infrastructure where the failure of a single component can bring down everything. This could be a single switch, firewall or Ethernet connection, storage not confgured as a RAID array or a solitary machine offerng any critical service.

Engineering around SPOFs is sometimes simple, such as buying an extra hard drive to use in a RAID 5 array, but at other times it can be expensive to resolve, such as buying a second switch and equipping all your servers with a redundant Ethernet card (configured in a bond). Although your engineering instincts should tell you to eradicate SPOFs wherever they exist, whether they're dealt with or not is often down to the manager and how much they care about their engineer's sleep.



servers two different names in the DNS, you'd assign the same name to both IP addresses. For example, a zone file might have the following entries:

```
www    IN A    192.168.56.10
www    IN A    192.168.56.12
```

If you're using the BIND DNS server, by default it deals with multiple entries such as this in a very particular way: each request for the IP of the www server will receive a different IP in response. So, Graham first looks up the address for **www.mycompany.com** and is given 192.168.56.10 as the first address to try; Andrew then requests the address, and gets 192.168.56.12; when Ben requests the address, he'll get 192.168.56.10 again.

This 'round-robin' response means that each server will deal with requests for an even fraction of your users. If one of your servers goes down, at least it's effecting only 50% of your users! If you had three servers, it would be only 33.33% of your users, etc.

This is not perfect, however. For starters, if a server goes down, some proportion of your users will still be effected. What's more, there's no requirement for web browsers or other clients to always visit the IP address first returned by the DNS server, so this technique may not even work! So while it's simple, it's probably better for distributing load than it is achieving HA.

HAProxy

A far better solution is to use a separate server as a load balancer. Dedicated load balancers ensure traffic is directed to the exact server you want, and they can perform health checks, ensuring traffic never gets sent to a broken server.

Load balancers work by sitting transparently between the client and the server. Rather than setting the DNS record for **www.mycompany.com** to the IP address of either web server, we configure the DNS to point at the IP address of the load balancer. All traffic will go to the load balancer instead, »

» that will look at each packet it receives and figure out which web server should receive and respond to this particular one.

Load balancers can take in to account things such as source IP address, destination port, "health" of the servers to which it can forward the traffic and many other things.

Of course, you can spend a lot of money on a commercial load balancer, but there's an excellent open source option called *HAProxy*. To get *HAProxy* working with a two-web server configuration, the first thing you need to do is install it. On most distributions, this is as easy as using your distribution's default package manager; on some, such as CentOS, you'll first need to install an extra repository, such as the Extra Packages for Enterprise Linux (EPEL) repository.

Once that's done, there's one configuration file that you must edit in order to get it working with your two web servers: **/etc/haproxy.conf**. As a very basic example, you might set up the following configuration:

```
frontend web_frontend *:80
  default_backend web_backend
backend web_backend
  mode http
  balance roundrobin
  server www1 192.168.56.10:80 check
  server www2 192.168.56.11:80 check
```

HAProxy

Configuration Manual

version 1.4.24-9

willy tarreau

2013/08/17

This document covers the configuration language as implemented in the version specified above. It does not provide any hint, example or advice. For such documentation, please refer to the Reference Manual or the Architecture Manual. The summary below is meant to help you search sections by name and navigate through the document.

› ***HAProxy* is the free/open source software of choice for those looking to achieve guaranteed high availability.**

Let's break down this sample configuration:

The frontend section tells the *HAProxy* server which port and address to listen on for incoming connections, and to which group of servers these connections should be forwarded. It's introduced by the **frontend** keyword, is then given a name, **web_frontend**, before the IP address and port is listed, in this case all IPs on port 80.

The **default_backend** parameter in the body of the section specifies the named group of backend servers to forward connections to.

The backend

The backend section is similar: it's introduced with the **backend** keyword, given a name, then a set of parameters and values are set in the body. In our example, we've said that this set of servers is running in HTTP mode (*HAProxy* can also work as a standard TCP load balancer), that the round-robin algorithm should be used for distributing requests, and then we go on to define our two backend nodes with the **server** parameter.

In the server statement, we've also specified the **check** option, which tells *HAProxy*, by default, to connect to the web server on the address and port specified and see if it gets a successful response, such as a HTTP 200 OK message. If it doesn't, it will stop sending new connections to that server, until it passes the test again.

HAProxy has a wealth of configuration options you can put in to this file, and we obviously don't have space to document them all here – it's better to look at their website: **http://haproxy.1wt.eu/download/1.5/doc/configuration.txt**. Once that's done, you should be able to start *HAProxy*,

RAID

Even if you've engineered your infrastructure to avoid single points of failure, life becomes easier if you can avoid ever having a server failure in the first place.

One of the most common causes of failure in servers are hard drives. They're mechanical devices and have a limited life – all hard drives will fail at some point. To mitigate the risk of a hard drive failure, you can configure your storage in a RAID (Redundant Array of Independent Disks) array.

RAID allows you to spread your data across multiple disks, with different data arrangements available depending on whether you're looking to boost performance or boost reliability. These configurations are known as 'levels', and are as follows:

» **RAID 0** Stripes data across drives. This improves performance, as reads and writes can be done in parallel across multiple drives, but a single failed disk still causes the loss of the entire array.

» **RAID 1** Mirrors data across drives. No performance gain, but if you lose one side of the mirror, the other keeps working, and so does your server. If you want to store 20GB, you'll need 40GB of storage.

» **RAID 5** Distributes data across drives, along with extra metadata, called 'parity' information. If a single drive fails, this parity information can be used in place of the lost data, but performance is less. Once the failed drive is replaced, the parity information can be used to rebuild the lost data on to the new drive, eventually restoring performance. If more than one disk fails, even while data is rebuilding on to the replacement drive, the parity information will be useless.

» **RAID 6** Like RAID 5, but supports the loss of two disks. If it will take a while to replace a disk, or if the drives are large (and hence rebuild times are long), this represents a better choice than RAID 5.

» **RAID 10** A mirrored stripe. That is, a mirror of a set of striped drives. This level attempts to deliver the performance benefits of RAID 0 and the redundancy of RAID 1.

RAID 5 and 6 are the most common when data integrity and availability are important. They have a lower storage overhead than RAID 1 or RAID 10 and still offer some read performance improvements. That said, they are slower writing data (because they have to calculate and write parity information), so you need to think carefully about which level you choose, based on your application's needs.

Once you've researched the levels carefully, you need to implement it. There are two main ways to do this: buy a 'hardware' RAID controller card, which manages the array, calculates parity, etc, and exposes the disks to the operating system as though they were a single disk. These often come with large caches, which can help in making the array seem faster than it otherwise would.

Of course, hardware controllers can be expensive, and so you might opt for software RAID. In Linux, software RAID is managed with the *mdadm* (multiple disk administration) utility. The syntax is quite simple:

```
mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sda /dev/sdb /dev/sdc
```

This example creates a new RAID 5 array, exposed as **/dev/md1**, made up of disks sda, sdb and sdc.

using the **service** command or **/etc/init.d script**. When you connect to the www domain, you should then find your connections being forwarded to one of the two web servers.

HA load balancers

We've made real progress by introducing *HAProxy* in to the setup, at least compared with no redundancy or round-robin DNS; the question that you should now be asking yourself, however, is what happens if the *HAProxy* server goes away? You'll have two working but useless web servers, since there's nothing to forward any traffic to them.

Initially, thinking about this kind of problem could lead you to a 'turtles all the way down' moment – that is to say, how can you introduce HA to your load balancer layer without adding yet another layer?

The simplest answer is to set up a 'virtual IP' address that gets shared between a pair of load balancers. That's right, along with your pair of web servers, you'll need to run a pair of load balancers! This might be starting to sound a bit wasteful to you – in order to serve traffic from a single web server, we've now introduced three further servers – a backup web server, a load balancer to manage the traffic, and a backup web server. Because of the extra resources required, you ought to think carefully about which HA technique you use.

If you're simply after a HA setup for a low traffic service, then a load balancer is overkill and you'd do better to skip the load balancer layer and jump straight to the virtual IP technique we're about to explain. If you're wanting HA and load balancing between an entire cluster of servers, however, then a separate load balancer with a virtual IP is perhaps the best solution for you.

Virtual IPs

With that caveat in mind, let's take a look at what a virtual IP is and how you can go about setting one up.

The basic idea of a virtual IP is that, as well as your two servers having their usual IP address, a third IP address is managed by software running on each server, which between themselves ensure that it's 'up' on only one of the two, at any one time.

For instance, as well as our servers having 192.168.56.10 and 192.168.56.11, they'll share the third address – 192.168.56.12. We usually configure one as the 'master', primary server and the other as the backup.

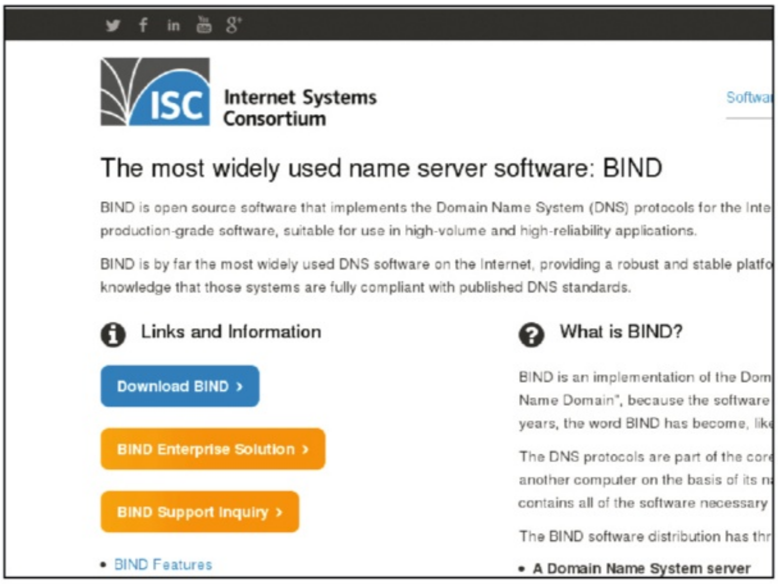
In normal operation, the backup server will keep checking that it can see the master server; essentially, that it can ping it and get a response. As soon as the backup server can no longer get a response, it will bring the configured virtual IP up on itself. If the master server re-appears at some point in the future, it will drop the virtual IP and the master server will bring it back up on itself.

To take advantage of this, you configure all your infrastructure, such as firewalls, to direct traffic destined for the HA resource, eg our web servers, to the virtual IP address. If the master fails, traffic will then automatically be redirected to the backup box.

Keepalived

As ever, there are many different tools you can choose to implement a virtual IP with, but the simplest one is *keepalived*. To set it up, install it and then put something similar to the following in **/etc/keepalived/keepalived.conf** on both servers:

```
vrrp_instance VI_1 {
  interface eth0
   state MASTER
   priority 101 # 100 on the backup box
   virtual_router_id 51

   virtual_ipaddress {
     192.168.56.12
   }
}
```



Internet Systems Consortium

Softwa

The most widely used name server software: BIND

BIND is open source software that implements the Domain Name System (DNS) protocols for the Inte production-grade software, suitable for use in high-volume and high-reliability applications.

BIND is by far the most widely used DNS software on the Internet, providing a robust and stable platfo knowledge that those systems are fully compliant with published DNS standards.

Links and Information

- BIND Features

What is BIND?

BIND is an implementation of the Dom Name Domain", because the software years, the word BIND has become, like

The DNS protocols are part of the core another computer on the basis of its n contains all of the software necessary

The BIND software distribution has thr

- A Domain Name System server

While most people know *Bind* for its ability to resolve names to IP mappings, it can also be configured to do simple load balancing. It's not as robust as the other options covered here, however, so consider your use of it carefully.

This configures a 'virtual router redundancy protocol' (VRRP) instance VI_1 with virtual IP address 192.168.56.12. If the machine with the greater priority stops responding, the virtual IP will be brought up on the machine that has the lower priority.

Keepalived can be extended with 'check scripts', which can be used to monitor the health of a service on a web server. So if you're using *keepalived* to maintain a HA load balancer, you could use the following in **keepalived.conf** to check whether the *HAProxy* process is running:

```
vrrp_script chk_haproxy {
    script "killall -0 haproxy"
    interval 2
    weight 2
}
```

This scrit will check if *HAProxy* is running every two seconds, and if it is, will add two to the priority. If you have this script on both machines, both will have two greater priority while *HAProxy* is running, and so the normal failover rules will apply. If *HAProxy* fails on the primary machine, however, it will now have one less priority than the backup, so the VIP will flip.

More to think about

Depending on the service that you're configuring for HA, there may be further techniques that you need to consider, each of which is specific to the different services. In past *LXFs*, for instance, we have already introduced you to *MySQL* replication, but you might also need to consider tools such as *DRBD* to replicate entire filesystems, slave DNS servers and others besides.

There are far too many possibilities to cover in a single article, but many of these specific techniques must also be coupled with load balancing or a virtual IP of one form or another. With a bit of luck, then, this article will have introduced you to the key concepts that will allow you to build an HA infrastructure. LXF

System performance Reclaim space on your Linux box from the command line

Disk usage:

Cached files from web browsing and software updates can clog up your computer. **Neil Bothwick** shows you how to clear up the mess.

Our expert

Neil Bothwick has a computer in every room, but won't disclose the location of his central server for security reasons.

Hard disks are big, cheap and fast these days, but no matter how much free space you think you have, it seems to disappear. Yes, there are all those ISO images and movies you downloaded, and maybe you have been trying out distros in *VirtualBox*, but there are other files that eat away at your free space, files you didn't put there and maybe don't even know about. So what are these files, where are they and how can we safely get rid of them?

The most common cause of hidden disk usage is caches. When a program accesses a file over the network, it often stores it locally before acting on it. This is usually for the purposes of caching, keeping a local copy so it can be accessed that much faster the next time you use it. Web browsers do this a lot, keeping copies of images from web pages so that when you go to the next page on the site they don't have to download the site's standard images again. Another reason is to check the file's integrity – package managers download software packages to a cache directory, then check that the file's checksum or signature is correct before trying to install from it. This protects against corruption in transit and, more importantly, malicious tampering with software packages. So where is all this space going? The **df** (disk free) command shows you how much space on a filesystem is used and available:

```
df -h /home
```

But how do you know where it is being used? There are some graphical options, which we'll come to later, but the standard command for this job is **du** (disk usage).

```
du -sch ~
```

This will show the total usage of your home directory. The **-h** turns on reporting in human-readable units, **-s** summarises each directory instead of listing all the contents and **-c** gives a total at the end. You can break down the usage further with

```
du -sch ~/*
```

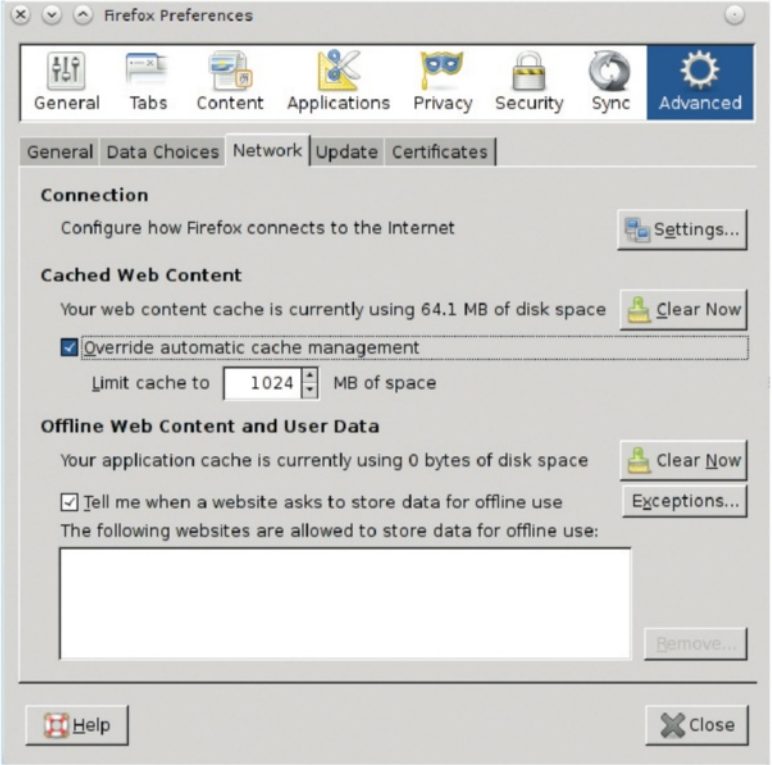
If you compare the total disk usage reported by those two commands, you will see a significant difference. This is because of the way the ***** wildcard operates in the shell; it does not match hidden files or directories, and that will include locations such as **.cache** and **.config**. To make *Bash* have ***** match dotfiles in the shell, turn on the **dotglob** option.

```
shopt -s dotglob
```

Zsh users can use ***(D)** as the pattern to math dotfiles. This now gives us a list of directories and files, showing how much disk space each uses, but we need to pick out the largest, which we do by piping the output of **du** to **sort**.

```
du -s -BM * | sort -n | tail
```

The **-BM** option to **du** lists all sizes in megabytes, the **-n** option for **sort** sorts numerically and **tail** shows us the ten greediest space users. You can repeat the command on any of these to see how their usage breaks down.



Connection

Configure how Firefox connects to the Internet

Cached Web Content

Your web content cache is currently using 64.1 MB of disk space

☑ Override automatic cache management

Limit cache to 1024 MB of space

Offline Web Content and User Data

Your application cache is currently using 0 bytes of disk space

☑ Tell me when a website asks to store data for offline use

The following websites are allowed to store data for offline use:

› ***Firefox* lets you limit space used for caches. The default is to decide itself.**

Browser bloat

Web browsing has long been established as a highly effective way of wasting time, but it can also consume far more disk space than you intend. Images displayed are also cached, which makes a lot of sense, but do you really want to keep images from sites you visited a year ago and are unlikely to ever return to? If you use *Firefox*, you can limit the cache size. Go to Preferences>Advanced>Network, tick the Override automatic cache management box and you can set a maximum size for the cache. *Chrome/ Chromium* users are less fortunate – the only way to limit the cache size is to start the browser with the **--disk-cache-size** option, which takes a value in megabytes.

Web browsers are not the only programs that can use large amounts of cache space. Many image viewers and file managers have a thumbnail view, and those thumbnails are cached. While it makes sense to have these available instead of generating them each time, do you still need all the thumbnails of your holiday snaps from five years ago?

When you list the contents of a directory, you see a time and date for each file, which is when it was last modified. Linux filesystems also store an atime for each file, the time it was last accessed. Some distros disable atimes because it has a slight effect on the speed of reading files, particularly small ones, but it is useful here. Those cached images may have been created years ago, but you want to remove only the

Free it up

ones you no longer use, and this is where atime helps. Using **find**, we can list all files that have not been looked at for, for example, 90 days

```
find .cache -type f -atime +90
```

We can remove these files with **find**, too

```
find .cache -type f -atime +90 -exec rm "{}" +
```

This depends on your filesystem having atimes enabled, which you can tell by looking at the output from the **mount** command. If it does not show noatime in the list of options for the filesystem in question, you can use this, otherwise you can use only modification times. The **find** syntax is the same, except you use **-mtime** instead of **-atime**.

If you use IMAP to read your emails, your mail program will cache a copy of each mail as you read it. Over time, these can build up, even mails that you are unlikely to read again. Bear in mind, though, that deleting cached mails will cause them to be downloaded again when you next use your mailer's search functions, so only clear this cache if space is really tight (or you never search your emails).

It's not only home

Apart from the caches in your home directory, your package manager may be using up more space each time you install updates or new software. Where these files are kept, and how you safely remove them, depends on the package manager. All distros based on Debian or Ubuntu use the *APT* manager (programs such as software centres and *Synaptic* are simply friendly front ends to *APT*). *APT* keeps copies of downloaded packages in **/var/cache/apt**. There is no need to delete them manually, *APT* provides options for cleaning up the cache. On a well used and frequently updated system, this can account for a sizeable chunk of disk space. The main command for this is **clean**:

```
apt-get clean
```

which removes all downloaded packages. There is also **autoclean**, which removes only those packages that are no longer available in the repositories, so it cleans out the obsolete stuff. Either of these can be run with the **--dry-run** option, to see what would be deleted without doing so.

RPM also stores downloaded packages in **/var/cache**, although the exact location depends on the package manager. *Yum* on Fedora uses something like **/var/cache/yum/x86_64/19/fedora/packages**, although the exact path depends on the version you are using. Sorting them by release version means you can easily dispose of any old packages left over after an upgrade from a previous version. The *Yum* equivalent of the above **apt-get** command is

```
yum clean all
```

This removes all downloaded package files from the cache, along with other cached data. The *RPM* database is not affected. There is also

```
yum clean packages
```

which removes files for packages that are no longer installed.

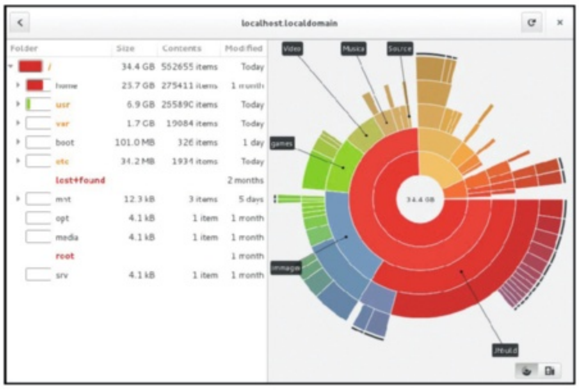
The default package manager on openSUSE is *zypp*, which keeps files in **/var/cache/zypp** but, unlike other distros, it defaults to deleting the files once it has installed from them, so no action is needed. Gentoo users can use **eclean-dist** to empty the **distfiles** directory. All the commands in this section need to be run as either the root user or with **sudo**.

So far, we have used command line programs. There are several reasons for this: they are all found on all distros (most of them are part of the **coreutils** package), they can be used even if a full disk prevents your desktop from loading, and they can be used remotely. There are also graphical programs we can use, and they are particularly useful for visualising your space usage. *Filelight* is a KDE program that shows the space usage of files and directories as a sort of nested pie chart, allowing you to move through the hierarchy finding the space wasters and opening a file manager window at the ones you want to examine more closely. Yes, there is a KDE program without a K in its name. The GNOME equivalent is called *baobab* although, thankfully, it is usually referred to as the *GNOME Disk Usage Analyser*.

These two programs are the graphical equivalent of **du**, there is also at least one program for removing files and freeing up space – *BleachBit* (**http://bleachbit.sourceforge.net**). *BleachBit* works on your home directory and knows about the types of files used by various applications and desktop environments. You can go through the list, ticking off the areas you want cleared, then hit Preview to see how much space it would save. A quick run here showed more than 2GB from my home directory alone.

Don't be too vigorous here, clearing browser caches may save a lot of space, but clearing all cookies and saved passwords will mean you need to log in again at each such site you visit. The Preferences window contains some useful options, such as a whitelist of directories to never try to clean and even an option to overwrite files before deleting them. However, the really paranoid among you (and that is most of us these days) would probably prefer to use **shred** to really get rid of the contents of files. Note that *BleachBit* is intended for cleaning out your home directory. Things such as package manager caches are still best handled from a terminal.

Now that you have freed up all that space, you can go and download a load more stuff to fill it up again! LXF



localhost.localdomain

Folder	Size	Contents	Modified
/	34.4 GB	552655 items	Today
home	25.7 GB	275411 items	1 month
usr	6.9 GB	255890 items	Today
var	1.7 GB	19084 items	Today
boot	101.0 MB	326 items	1 day
etc	34.2 MB	1934 items	Today
lost+found			2 months
mnt	12.3 kB	3 items	5 days
opt	4.1 kB	1 item	1 month
media	4.1 kB	1 item	1 month
root			1 month
srv	4.1 kB	1 item	1 month

› *Filelight* and *GNOME Disk Usage Analyser* give a graphical representation of disk usage.

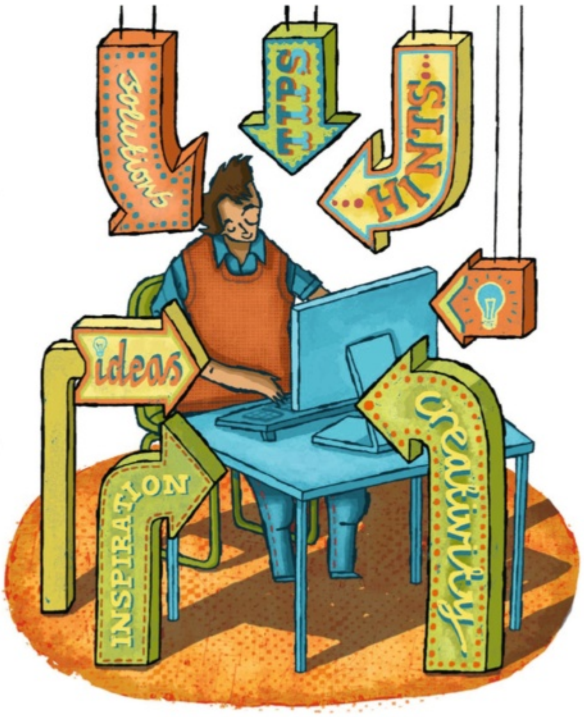
Desktop environment Create a custom solution from individual parts

Desktops: Build

Yes, you can create a highly customised, low-resource desktop environment without a single line of source code. **Mike Saunders** explains how...

Our expert

Mike Saunders started using Linux with FVWM (Red Hat 5.1 days), spent a few years with Window Maker, and now feels at home in



Titlebars, launchers, panels, notification areas, background images – these are all things we expect in any modern desktop environment. Usually, these features are provided by individual programs, so if you're running *Xfce* and enter **ps ax** in a terminal, for instance, you'll see **xfwm** for the window manager, **xfce4-panel** to provide on-screen panels, and so forth. If you start terminating these programs, bits of your desktop environment will start to disappear from the screen. KDE and Gnome work in a very similar fashion.

Now, this is all good and well, but we're Linux geeks, right? The defaults do an adequate job, but we run this OS so that we can customise it to our hearts' content, rip out bits we don't like, and explore the inner workings of the system. KDE, Gnome and *Xfce* look like hugely complicated beasts, but they're really just a series of programs started in sequence. You can replace these programs individually, or you can choose your own combination of window manager, panel, workspace switcher and so forth – and that's exactly what we're going to do in this tutorial.

There are many lightweight window managers out there that offer the bare minimum of functionality, so other developers have written the extra components you need to create a fully-fledged desktop environment. We'll explore the different choices available, and then show you how to piece them all together into a custom desktop made for you. If you're running KDE, Gnome or *Xfce*, you'll find that the result is a much snappier and less memory-intensive experience. But most importantly, when your fellow Linux users are talking about upgrading to the latest KDE or Gnome, you can wave your hand and say: "Pfft! Those are old-hat. I made my own desktop environment last night, and I'll install it for you if you like." Geek points are coming your way...

Choose a window manager

By far the most important component is the window manager. This (normally) provides titlebars so that you can move windows around on the screen, along with buttons to perform actions on the window – maximise, close, etc. The WM also (normally) adds handles to the edges of windows for resizing. Without a WM, your applications could still run, but you wouldn't be able to move them around or resize them. There are some novel window managers that take different approaches, such as automatically arranging windows into grids (tiling WMs), and they don't need titlebars.

Here, we'll describe some of the best choices and explain how to use them. If you install them via your distro's package manager, you should be able to log out of your current session and choose them in your login screen. If you don't see an entry for your WM of choice, try selecting a fallback/terminal option in the login screen; some distros provide this. You'll see an empty terminal window and you can enter the appropriate command to start the WM, as described below. If you can't get the WM to start at all, worry not: we'll show you how to create a new login screen session later on.

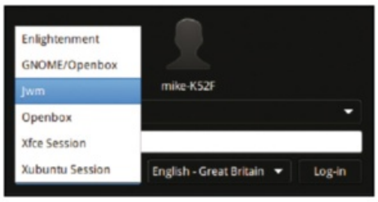
» Name: *JWM*
» Website: http://joewing.net/projects/jwm/
» Command to start: jwm

JWM is very bare-bones, being written in C and only making use of the core X libraries as dependencies. Consequently, it's screamingly fast. *JWM* has a fairly usable default panel, containing a program menu, workspace switcher and taskbar, but you'll probably want a more featureful panel from the options we'll look at later.

By default, the program menu (also available by left-clicking on the desktop) contains only entries to start a terminal, lock the screen and log out – you can customise this by editing **/etc/jwm/system.jwmrc** (or **/usr/local/etc/jwm/system.jwmrc** if you built it from source).

This XML-based configuration file is easy to understand, and via it you can also customise the WM's fonts and keybindings.

› Adding window managers via your package manager should create new login screen sessions – but we'll also show you how that works.



your own

Name: *Openbox*
Website: www.openbox.org
Command to start: openbox

Openbox is part of the *box family of window managers, which includes *Fluxbox* and *Blackbox* (the original, from which the others are derived). It's perhaps the most popular low-fat window manager, and is used in the LXDE desktop. *Openbox*'s default configuration is as bare-bones as it gets: as soon as you start it, you'll see a grand total of nothing. Just your mouse pointer in the middle of the screen.

Right-click, however, and a small menu will pop up, from which you can open a terminal window, launch a web browser, or run the configuration tool. It's this tool that makes *Openbox* our favourite choice for custom desktop environments, because you can configure almost everything in the window manager without having to poke around inside text files. Plus, while *Openbox* is very limited on the feature front, it's highly standards-compliant and interacts excellently with other tools.

The keyboard shortcuts are typical: Alt+Tab to cycle through windows, Alt+F4 to close programs, and Ctrl+Alt+Left/Right cursor keys to switch between different virtual workspaces.

Name: *Ratpoison*
Website: http://ratpoison.wxcvbn.org
Command to start: ratpoison

Here's a very different kind of window manager: it's built around a concept called "tiling". Instead of the usual system of draggable windows that can overlap one another, in *Ratpoison* windows are allocated areas on the screen. It takes a while to get used to, but as you learn the keybindings used to manage windows, you become less and less dependent on the mouse (hence the name).

On first start, *Ratpoison* is completely bare, and you'll find that the mouse does nothing. Hit Ctrl+t followed by "!", though, and a tiny box will appear in the top-right corner, which lets you enter a command (eg, **xterm**). When you launch a program, by default it will appear in full-screen mode, and once you have multiple windows open, you can switch between them Using Ctrl+t, followed by n and p (next and previous). Enter Ctrl+t w to show a numbered list of windows and Ctrl+t followed by a number to switch to a specific window. Now, the real fun begins in tiling mode. With a few windows open, hit Ctrl+t s to split the screen vertically into two zones, called frames. The current window will appear in the top frame, while the previous window will appear underneath. Hitting Ctrl+t, followed by S (capital letter) splits the screen horizontally. This is excellent if you have a large and high-resolution monitor, as you can, for instance, create a large frame for your web browser, and have a couple of smaller terminal windows sitting next to it.

To switch between frames, hit Ctrl+t, followed by the cursor keys (eg, the down key to switch to the frame underneath). You can make a frame full-screen again with Ctrl+t Q. For a full list of keybindings, enter Ctrl+t ?, and visit the *Ratpoison* wiki at **http://tinyurl.com/obcarv5** for details on more features in the WM.

Other window managers worth looking at include *Sawfish* (**http://sawfish.wikia.com/wiki/**), *WindowLab* (**http://nickgravgaard.com/windowlab/**) and *Pekwm* (**www.pekwm.org**).

Talking the same language

It's no coincidence that all these tools can work together. Critics accuse the Linux desktop of being a mish-mash of old and new technologies, with little binding them together, but that's not fair. Sure, there's a lot of old code still around, but programs work together thanks to standards.

EWMH, the Extended Window Manager Hints specification, "defines interactions between window managers, applications, and the utilities that form part of a desktop environment". In other words, it lets these components share information, providing a more cohesive experience. A good example is workspace switchers: they often show miniature thumbnails of the workspaces and the windows in them. Usually, you want only to see application windows in these thumbnails, and not the usual desktop furniture. Thanks to EWMH, a panel or a window manager can tell a pager that a window isn't meant to be displayed in the thumbnails. See **http://standards.freedesktop.org/wm-spec/wm-spec-latest.html** for the full spec – it's a long read, but essential if you fancy writing your own WM.

Choose a panel

So, we have the job of managing windows sorted out. Now we want a panel for launching programs, switching between windows and workspaces, showing a clock and so forth. As mentioned, *JWM* includes a simple panel, but we'll look at some more featureful options here.

Name: *Docky*
Website: www.go-docky.com
Command to start: docky

We absolutely love the description on *Docky*'s website: "If there ever was something made of pure awesome in the history of mankind, then its name is *Docky*". And, you know, we can't really disagree – it's very cool indeed. *Docky* is fairly heavy for a panel, with a lot of dependencies, so if you're creating a custom desktop for low-memory machines it's best to try the other options. You should be able to find *Docky* in most distros' package repositories (it's just a **sudo apt-get install docky** away in Ubuntu-based distros), and when you start it, you'll see a small black panel at the bottom of the

» screen. There's a single static icon that shows an anchor – this opens up the Settings dialog. From here, you can choose Panel mode, which extends *Docky* to fill the width of the screen, and change the theme (some of them are truly gorgeous). Switch to the Docklets tab of the Settings dialog to add other static items to the panel, such as a clock, workspace switcher and battery monitor.

Docky doesn't include a main menu, which is a bit limiting, but if you start programs via your window manager you'll see icons for them appear on the panel. Right-click on an icon and choose "Pin to dock" to make the icon remain in the panel, even when you close the program. In this way, you can populate your panel with your most-used programs, so the lack of a main menu isn't a huge deal.

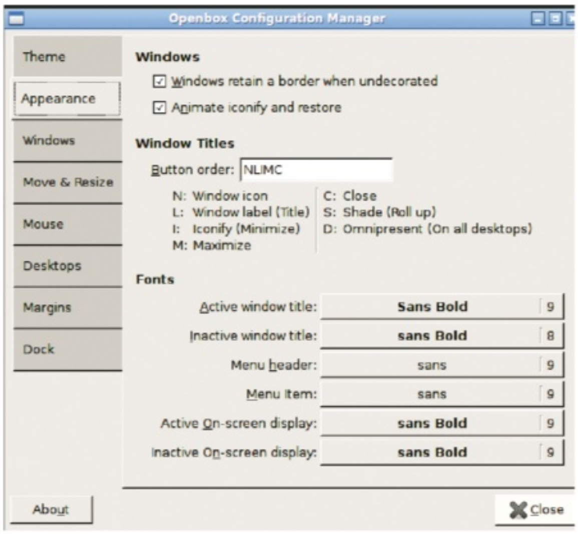
» **Name:** *Cairo-Dock*
» **Website: http://glx-dock.org**
» **Command to start: cairo-dock**

If you want eye candy, *Cairo-Dock* delivers it in spades. Its "3D Plane" view is strikingly similar to Mac OS X's dock, with lush reflection effects and patterned lines separating different areas. Icons grow in size as you run the mouse pointer over them, and if you hover over a folder, a preview of its contents springs up onto the screen. The whole thing is really beautiful – but it requires some oomph from your graphics card, of course.

Unlike *Docky*, *Cairo-Dock* does include a main menu based on system settings; in other words, it'll show the same items that you see in other window managers and desktop environments. You can right-click on any item to configure it, or go into Cairo-Dock > Configure to customise the panel itself. And it's immensely configurable, with plenty of add-ons to enable, keybindings to tweak and size and position settings to change.

» **Name:** *WBar*
» **Website: http://code.google.com/p/wbar/**
» **Command to start: wbar**

This is the lightest of the three panels covered here, and starts up almost instantly. It features a transparent background, animated growing icons when you mouse over them, and a graphical configuration tool (install **wbar-config** alongside the panel itself). By default, it's displayed vertically, but you can click on the top-most icon to change this, along with other settings.



Windows

☑ Windows retain a border when undecorated
☑ Animate iconify and restore

Window Titles

Button order: NLIMC

N: Window icon
L: Window label (Title)
I: Iconify (Minimize)
M: Maximize
C: Close
S: Shade (Roll up)
D: Omnipresent (On all desktops)

Fonts

Active window title: Sans Bold 9
Inactive window title: sans Bold 8
Menu header: sans 9
Menu Item: sans 9
Active On-screen display: sans Bold 9
Inactive On-screen display: sans Bold 9

› ***Openbox* has its own configuration tool (*obconf*), which obviates the need to fiddle with text files.**

Add and remove launchers via the Icons tab, and if you want the panel to act more like a taskbar, there's an option for this under the Preferences tab. To customise the zoom levels, transparency effects and distance between icons, see the Effects tab.

If none of these docks float your boat, try *tint2* (**http://code.google.com/p/tint2/**), *PerlPanel* (**http://savannah.nongnu.org/projects/perlpanel**) or the *Avant Window Navigator* (**https://launchpad.net/awn**). The last two are a bit dated now, but you can still find packages for various distros on the net.

Choose some extras

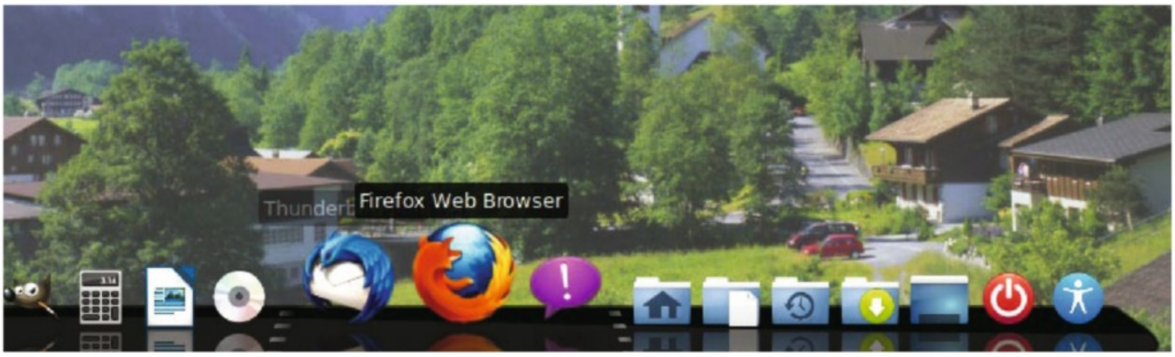
Most lightweight window managers don't include any way to set the desktop background image, so you'll need another tool for that. *hsetroot* is a simple little program that's available in most distros as a standalone package, or occasionally as part of *HackedBox* (another small WM). Once you have it installed, using it is as simple as:

```
hsetroot -full /path/to/image.jpg
```

That places the full image on the screen, without any changes. Other options include **-center**, **-tile** and **-fill** (to stretch it). You can also use PNG images and other formats.

If your choice of dock or panel doesn't include a virtual workspace switcher, you'll need to get a standalone one. There isn't a massive variety of choices in this area, but the one we recommend (because its dependencies are minimal and it works almost everywhere) is *NetWMPager*. Go to **http://sourceforge.net/projects/sf-xpaint/files/netwmpager/** and grab the latest source code (2.04 at the time of writing) and build it like this:

```
tar xfv netwmpager-2.04.tar.bz2
cd netwmpager-2.04/
```



› **Fancy adding some Mac OS X-esque spit shine to your custom desktop? *Cairo-Dock* is what you need.**

```
./configure
make
sudo make install
```

If you encounter an "undefined reference to symbol XGetWindowAttributes" problem after the **make** step, edit **config.mk** and change the **XFT_LIBS** line so it reads like this:

```
XFT_LIBS = -lXft -lX11
```

Save the file and run the last two steps again. Now you can start it with **netwmpager**, and you'll see a pager window showing desktops in the bottom-left corner.

By default, it's very small and auto-hides; to change this, edit **/usr/local/share/netwmpager/config-example** and modify the **geometry** and **auto_hide** lines. If you need a notification area (aka system tray), *Stalonetray* is a fine example. It's available in Ubuntu's package archives and also via source code at **http://stalonetray.sf.net** – if you're compiling it by hand, extract it and run the **./configure**, **make** and **sudo make install** steps as above.

Then launch it like so:

```
stalonetray -geometry 5x1+50+20
```

This creates a notification area five slots wide by one slot high, 50 pixels across and 20 down (from the top-left corner of the screen). You can, of course, make it any size and place it wherever you want.

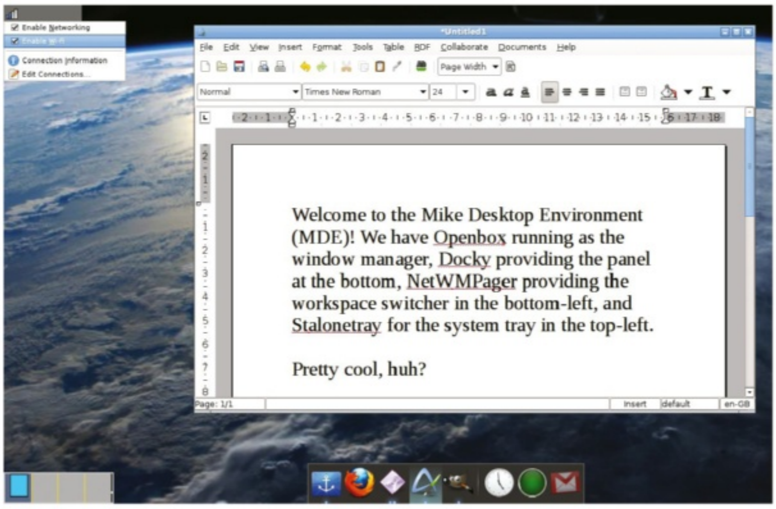
Put it all together

Now we're ready to put all of the components together, and make a desktop environment. You can use any combination of the previously mentioned programs; in our case, we'll use *Openbox*, *Docky*, *hsetroot*, *NetWMPager* and *Stalonetray*. The first thing to do is to create a script that launches the required programs. For our custom desktop, we're putting this in **/usr/local/bin/mikedesk-start**:

```
#!/bin/bash
docky &
hsetroot -fill /usr/share/backgrounds/space-02.jpg &
netwmpager &
stalonetray -geometry 5x1+0+0 &
openbox
```

The **&** after the commands here means "run them in the background" – ie, so they can all run at the same time, and don't wait for each other to finish. It's important that your window manager goes on the last line, without the **&**, otherwise the session will close immediately.

Save the file and make it executable (eg, **sudo chmod +x mikedesk-start**). The next step is to create a login screen session, so make a new file called something like **/usr/share/xsessions/mikedesk.session** with the following contents, customised for your setup:



Welcome to the Mike Desktop Environment (MDE)! We have Openbox running as the window manager, Docky providing the panel at the bottom, NetWMPager providing the workspace switcher in the bottom-left, and Stalonetray for the system tray in the top-left.

Pretty cool, huh?

And here's the result of our work: a desktop environment built with our own selection of components.

```
[Desktop Entry]
Name=MikeDesk
Comment=Mike Desktop Environment
Type=XSession
Exec=/usr/local/bin/mikedesk-start
TryExec=mikedesk-start
```

Save it, log out of your current session and you should see a new one in your login manager. Choose it, log in and enjoy the fruits of your labour! Congratulations, you are now officially awesome. LXF

Help! My Gtk and Qt apps look rubbish

This, unfortunately, is a common problem when switching to another window manager. You set up everything perfectly, install a great theme, choose a dazzling desktop wallpaper, and then... Your Gtk and Qt applications look like something out of the 1980s. Flat menus, blocky widgets, dark grey backgrounds and ugly fonts – it's almost like a joke.

Gtk and Qt can no longer determine which theme to use, so they go with the ugly built-in defaults. If you're running Gnome or *Xfce*, Gtk apps can ask the desktop to provide theme information, but with a standalone window manager these toolkits need to rely on configuration files. For Gtk, there are a few options; it's best to try a combination. *gtk-chtheme* lets you set the theme used by Gtk 2.x applications, but not 3.0. For the latter, edit **~/.config/gtk-3.0/settings.ini** so that it specifies the theme name:

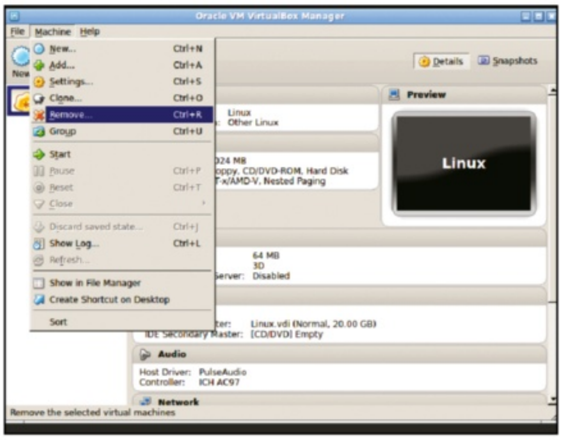
```
[Settings]
gtk-theme-name = Clearlooks
gtk-fallback-icon-theme = gnome
```

If neither of these have the desired effect and you have Gnome installed, try running **gnome-settings-daemon**. That will re-theme running Gtk apps using the settings specified in Gnome, but it will also reset your desktop wallpaper.

For Qt, you have two options: to force Qt apps to use Gtk themes, edit **~/.config/Trolltech.conf** and add this:

```
style=GTK+
```

Alternatively, install the **qt4-qtconfig** package and then run **qtconfig**.



Linux
Other Linux
024 MB
oppy, CD/DVD-ROM, Hard Disk
T-x/AMD-V, Nested Paging
64 MB
3D
Server: Disabled
Linux.vdi (Normal, 20.00 GB)
IDE Secondary Master: [CD/DVD] Empty
Host Driver: PulseAudio
Controller: ICH AC97
Remove the selected virtual machines

Here's how *VirtualBox* (a Qt app) looks with Qt's default theme – it's like Windows 95, all over again.

Drupal: Instant

Shashank Sharma discovers a faster way of creating *Drupal* websites. Is *Drupal Distributions* everything he believes or is he just easily excitable?

Our expert

Shashank Sharma has been writing about free software for more than four years, including the Linux Foundation. He is the co-author of *Beginning Fedora*.

The default *Drupal* installation includes a select bunch of modules and themes. But if these don't meet your needs you have to turn to *Drupal*'s thousands of modules to find the ones that offer the functionality you wish to provide. While there's nothing wrong with this approach, it can be time consuming. Depending on the type of website you wish to develop, it can take a week or longer to track, install and configure these additional modules and their dependencies. *Drupal Distributions* offers a happy alternative.

In addition to the *Drupal* core, a distribution includes all the modules and configuration needed to create a website. The advantage with using distributions is you don't have to track it down and configure all the modules needed for that type of website. You just need to download and install the distro, in the same way you would a regular *Drupal* release.

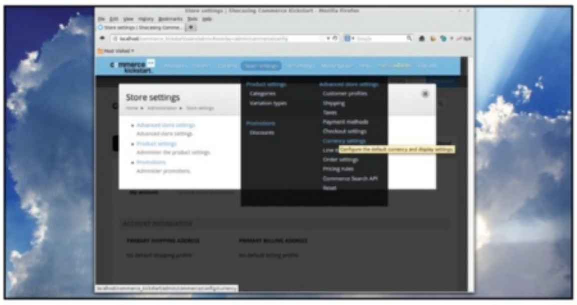
If you frequently find yourself retracing your steps for all your *Drupal* deployments, *Distributions* is just the thing for you. For instance, if you always install the same five modules as the first step, you can create a distribution that offers these modules installed and enabled. You can then install this distribution instead of the stock *Drupal* release to save time.

Deconstructing distributions

A distribution comprises the core *Drupal* files coupled with all the modules you need for a specific kind of website. Consider, as an example, the Kickstart distribution. In addition to the core *Drupal* files and the standard set of modules and themes, this distribution includes additional modules and themes to help you create your own commercial website. Once installed, the distribution provides a working, ready to use commercial website, complete with a checkout kart, and everything else you'd expect from a commercial site.

The most important part of a distribution is the installation profile. You can think of it as the fundamental building block of a distribution. If you've installed *Drupal 7* in the past, you must have already come across two installation profiles. When installing *Drupal*, the installation wizard asks you to choose between a Standard or Minimal installation profile.

These describe the modules and themes that need to be activated and available for use once *Drupal* is installed. The installation profile is used to inform *Drupal* what modules, themes, and other settings you wish to use with the *Drupal* installation. It is important to understand the distinction between installation profiles and distributions.



› The navigation menu may not seem like an important feature but it can make or break distributions.

The installation profile does not come bundled with any modules and themes. It's only a list of all the settings and configuration you wish to use on your *Drupal* installation. For example, if you were to install a blogging installation profile, it would configure settings, such as content type, roles, permissions, etc, that would enable multiple users to create content on your site. But, if the installation profile relies on additional modules, such as a module for rating the content on the site, you will have to install these manually.

A distribution, on the other hand, includes the core *Drupal* files and the additional modules and themes. As wonderful as *Drupal* is, the simplicity of getting a working website out of the box, requiring no effort on your part, is the reason many news sites, government organisations and non-profit organisations are using distributions to build their websites.

Distinct-looking distributions

While two or more distributions may target the same audience, the sheer number of different *Drupal* modules guarantees that no two modules are exactly alike.

A distribution differs quite a bit from the standard *Drupal* release. We're not just talking about the added functionality, thanks to the additional modules that distributions ship with.

The distributions almost always have a non-standard administrative menu. There are a number of different site navigation modules available for *Drupal*, and the makers of a distribution usually choose one of these alternative modules instead of the traditional menu you get with the standard *Drupal* release. The choice of navigation module usually depends on the purpose of the distribution.

In fact, some distributions, such as Commerce Kickstart, create their own administrative menu. For such complex and feature-rich distributions, the standard navigation and administrative menu is highly impractical.

No matter how different a distribution seems, the administrative functions are the same. This means running Cron jobs, applying updates, managing modules, and the other administrative tasks *Drupal* lets you run, are done much the same as you would with the traditional *Drupal* release.

Now that we've covered the basics of distributions, it's time to create our own special-purpose *Drupal* distribution.

We've already mentioned how the most important part of a distribution is the installation profile. You can think of the installation profile as a fundamental building block. You cannot make a distribution without first creating the installation profile for your distribution.

The first thing you need to do is create a *Drupal* website that you wish to turn into a distribution. Make sure you install all the modules and themes you want for the site. Forget

websites

about adding content to the site, we'll create a distribution out of this installed *Drupal* website.

The installation profile essentially contains these three files: **profilename.info**, **profilename.install** and **profilename.profile**. The content of these files depends on what modules and features you wish to provide.

The **profilename.info** contains at the very least a name for the profile, a brief description of the profile, the core *Drupal* that the profile is built for and a list of modules that need to be enabled automatically.

```
name = Profile name goes here
description = Description of what the profile does.
core = 7.x
dependencies[] = views
dependencies[] = ctools
dependencies[] = advanced_menu
```

The stock *Drupal* release ships with **.info** files for the Standard and Minimal installation profiles. You will find them under the **profiles/** directory.

The **.install** file is a PHP script with instructions to insert content into the database during the installation process. The following is the **.install** file for the Minimal installation profile:

```
<?php
/**
* Implement hook_install().
*
* Perform actions to set up the site for this profile.
*/
function profilename_install() {
¬† include_once DRUPAL_ROOT . '/profiles/minimal/
minimal.install';
¬† minimal_install();
}
?>
```

Last, we have the **.profile** file. This is also a PHP file, but it's used when you wish to add steps to the installation process. So, if you want users to specify their current geographical location, or the political party they support, you have to add that to this **.profile** file.

There was a time when you would have had to spend a great amount of time, hours maybe, trying to write those files manually, even if you were well versed with *Drupal* internals and PHP. But those days are behind us now. The Profiler and Profiler Builder modules, coupled with *Drush*, can create these files for you in a matter of minutes.

Once you have your site up and running, you can use the

```
drush dl profiler
```

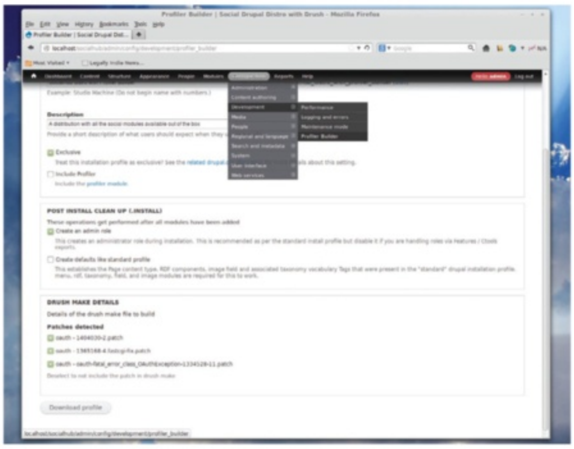
and

```
drush dl profiler_builder
```

commands to download the modules and then the

```
drush en profiler
```

and



Profiler Builder | Social Drupal Distro with Drush - Mozilla Firefox
Description
A distribution with all the social modules available out of the box
Exclusive
Include Profiler
Performance
Logging and errors
Maintenance mode
Profiler Builder

POST INSTALL CLEAN UP (.INSTALL)

These operations get performed after all modules have been added

☑ Create an admin role

□ Create defaults like standard profile

DRUSH MAKE DETAILS

Details of the drush make file to build

Patches detected

Download profile

› You can magically arrive at installation profiles with the Profiler Builder module.

Managing Drupal with Drush

The *Drupal* command-line shell, *Drush*, can make it easy for you to perform an internal clean-up, do a fresh *Drupal* install, install and enable modules, create users, and various other things, without once touching the mouse.

While most Linux distributions carry *Drush* in their repositories, it's best to install it using the official Pear channel. If you don't already have Pear installed, you can use your distribution's package management software. The command

```
sudo apt-get install php-pear
```

will do the job.

You should also install the **PHP5-mysql** and **PHP5-gd** packages. You're now ready to install *Drush*:

```
pear channel-discover pear.drush.org
pear install drush/drush
```

With *Drush* installed, you can now easily install *Drupal* and assorted modules from the command line itself. Make sure the directory is user-writeable and run the

```
drush dl drupal
```

command. This will download the latest *Drupal* release from the project website in the current directory. Change into the downloaded **drupal-7.23** directory.

You can now install *Drupal* with

```
drush si standard --db-url=mysql://
dbusername:dbpassword@localhost/dbname -
site-name="Drupal site with Drush"
```

This command is the equivalent of running through the browser-based installation wizard. We first specify the installation profile we wish to use (Standard in this case), and then give the details for the database that we've created for this website.

Now that *Drupal* is installed, you can install modules into this new site with the command

```
drush dl modulename command
drush en profiler_builder
```

commands to enable them. Our website, called Socialhub, is located under the **/opt/lampp/htdocs/socialhub** directory. Open a terminal and switch to this directory. Now run the

```
drush distro socialhub
```

command to create the installation profile. Make sure you have write permissions to this directory, or use

```
sudo
```

or

```
su
```

as needed.

```
$ cd /opt/lampp/htdocs/socialhub
$ drush distro socialhub
Wrote .tar file socialhub.tar to current directory
[ok]
$ tar tvf socialhub.tar
-rw-r--r-- 501/501      1360 2013-09-25 14:12 socialhub/socialhub.info
-rw-r--r-- 501/501      3609 2013-09-25 14:12 socialhub/socialhub.install
-rw-r--r-- 501/501       709 2013-09-25 14:12 socialhub/socialhub.profile
-rw-r--r-- 501/501      1623 2013-09-25 14:12 socialhub/drupal-org.make
-rw-r--r-- 501/501       498 2013-09-25 14:12 socialhub/local.make.example
```

The **drush distro** command generates the installation profile files and presents a TAR file. You can use the **tar tvf** command to view the contents of the TAR file. As you can see, the command generates the **.install**, **.profile** and **.info** files for us, along with a **.make** file, which we'll use to build our distribution. You can alternatively use the

```
drush distro socialhub --untar
```

command. This will create a directory called **socialhub** within the current directory, containing the installation profile files.

If you're uncomfortable on the command line, you can also create the installation profile graphically. Log in to your *Drupal* website as the administrator and click Configuration > Development > Profiler Builder.

Fill in all the details, such as profile name and description, and when you're done, click the Download profile button at the bottom of the page. Just like the

```
drush distro/
```

command, the Profiler Builder module, too, creates a TAR file containing the **.info**, **.profile** and other files.

While Profiler Builder is ideal for creating an installation profile, you can't use it to create a distribution that provides settings. The module will create only an installation profile that reflects all the modules and themes you've used to create your website. It doesn't retain the configuration.

This means when you install *Drupal* using your own distribution, you will get all the modules and themes out of the box, but you will still have to configure them.

If you want to provide the configuration for the modules as well, you have to use *Drupal*'s Features modules. This is the de facto method for exporting the settings and configuration used on your Drupal site.

Building distributions

You've already got a TAR file that contains the installation profile, along with the **drupal-org.make** file. It's this **.make** file that is used to create the distribution. Extract the files from the TAR file. Your **.make** file should look something like:

```
; social_drupal_distro_with_drush make file for d.o. usage
core = "7.x" api = "2" ; +++++ Modules +++++ projects[admin_menu][version] = "3.0-rc4"
projects[admin_menu][subdir] = "contrib"
projects[ctools][version] = "1.3"
projects[ctools][subdir] = "contrib"
projects[profiler_builder][version] = "1.0"
projects[profiler_builder][subdir] = "contrib"
projects[simplenews][version] = "1.0"
projects[simplenews][subdir] = "contrib"
projects[media][version] = "1.3"
projects[media][subdir] = "contrib"  ///END CODE///
```

This is only a partial snippet from our **make** file. You need to edit the file and add

```
projects[] = drupal
```

in a new line under the **api = "2"** line so that the file looks like:

```
; social_drupal_distro_with_drush make file for d.o. usage
core = "7.x"
api = "2"
projects[] = drupal
```

The rest of the file remains the same. As we will use *Drush* to build our distribution, this line tells *Drush* that *Drupal* is the main component of our distribution. We now use this **.make** file to generate the distribution with the

```
drush make --prepare-install --tar socialhub/drupal-org.make socialhub
```

command. This process may take some time, depending on the size of your website, which depends on the number of modules and themes.

When you run this command, *Drush* will download the standard *Drupal* release, along with all the additional modules and themes you've installed on your website and create a file called called **socialhub.tar.gz** in the current directory.

To test your distribution, uncompress the **.tar.gz** file. Next, create a new database and user. Finally, point your browser to where you extracted the distribution to begin the graphical installation. You can even offer your distribution to the world. But if you want to host it on *Drupal*'s website, you still have a way to go. Instead of the full package made with the

```
drush make --prepare-install
```

command, you need only the **.make** file to host the distribution on **Drupal.org**. The command

```
drush make --generate-makefile drual-org.make
```

will do this. Refer to **https://drupal.org/node/642116** for more details on how to package a distribution for **Drupal.org**.

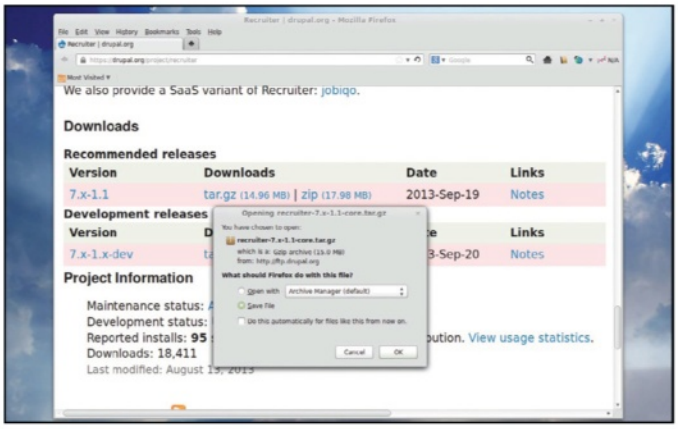
Custom modules

For the most part, a distribution is just a means of arriving at a more feature-rich *Drupal* installation, thanks to its bundled modules. You can apply updates to the modules as you would on a regular *Drupal* installation. In fact, most popular distributions provide regular updates for *Drupal* core and the additional modules included in the distribution.

But there are some distributions that are built around custom modules created by the developers themselves. When you use such distributions to build your *Drupal* website, you have no choice but to wait for the developer to release updates to the module and distribution. The downside of using such specially designed distributions is that you might get stuck with a *Drupal* site that is difficult to update should the developer ever stop maintaining the distribution.

› All additional modules for your distro are listed under sites/all/modules/contrib directory.

Installing a distribution



We also provide a SaaS variant of Recruiter: jobiqo.

Downloads

Recommended releases

Version	Downloads	Date	Links
7.x-1.1	tar.gz (14.96 MB) | zip (17.98 MB)	2013-Sep-19	Notes

Development releases

Version	D	e	Links
7.x-1.x-dev	ta	3-Sep-20	Notes

Project Information

Maintenance status:
Development status:
Reported installs: **95** … ution. View usage statistics.
Downloads: 18,411
Last modified: August 13, 2013

Opening recruiter-7.x-1.1-core.tar.gz

You have chosen to open:
recruiter-7.x-1.1-core.tar.gz
which is a: Gzip archive (15.0 MB)
from: http://ftp.drupal.org

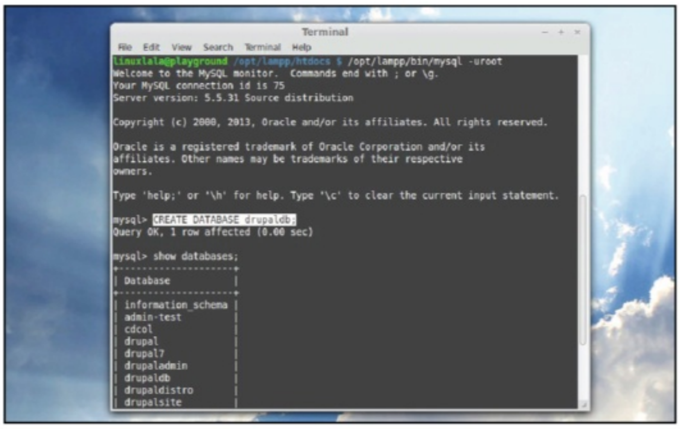
What should Firefox do with this file?

○ Open with Archive Manager (default)
● Save File
□ Do this automatically for files like this from now on.

Recruiter | drupal.org - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Cancel
OK

1 Download the distribution

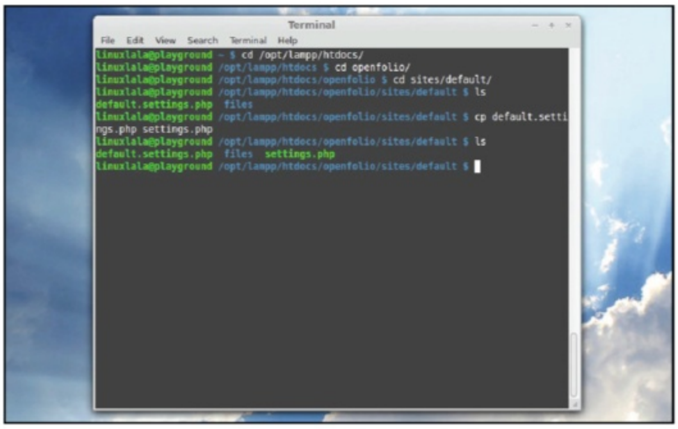
Download the distribution tarball from the project page and uncompress it into the target directory. This could be inside the **htdocs** directory or under **/var/www/** depending on your system setup. You may want to rename the uncompressed directory.



```
linuxlala@playground /opt/lampp/htdocs $ /opt/lampp/bin/mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 75
Server version: 5.5.31 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE drupaldb;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| admin-test         |
| cdcol              |
| drupal             |
| drupal7            |
| drupaladmin        |
| drupaldb           |
| drupaldistro       |
| drupalsite         |
```

2 Create a database and user

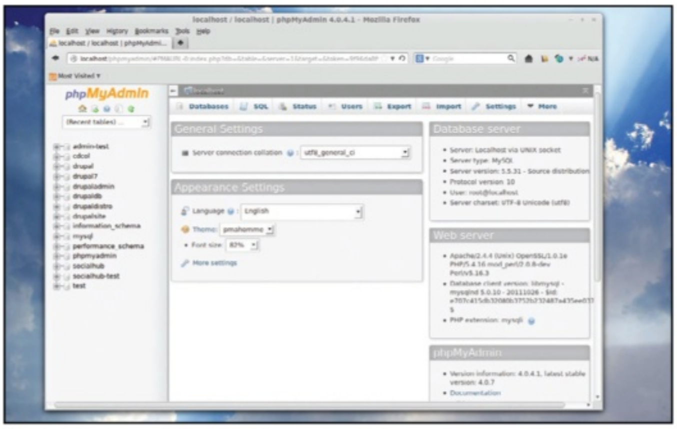
You can use the *MySQL* prompt or *PHPMyAdmin* to create the database and user. Keep the username and database handy, as *Drupal* will ask for these during the installation. You have to do this even when installing the standard *Drupal* release.



```
linuxlala@playground ~ $ cd /opt/lampp/htdocs/
linuxlala@playground /opt/lampp/htdocs $ cd openfolio/
linuxlala@playground /opt/lampp/htdocs/openfolio $ cd sites/default/
linuxlala@playground /opt/lampp/htdocs/openfolio/sites/default $ ls
default.settings.php  files
linuxlala@playground /opt/lampp/htdocs/openfolio/sites/default $ cp default.setti
ngs.php settings.php
linuxlala@playground /opt/lampp/htdocs/openfolio/sites/default $ ls
default.settings.php  files  settings.php
linuxlala@playground /opt/lampp/htdocs/openfolio/sites/default $ 
```

3 Rename default.settings.php

Navigate to the **sites/default** directory within the newly uncompressed directory and rename the **default.settings.php** file to **settings.php**. Point your browser to the uncompressed directory and proceed with the graphical installation.



General Settings
Server connection collation: utf8_general_ci

Appearance Settings
Language: English
Theme: pmahomme
Font size: 82%
More settings

Database server
- Server: Localhost via UNIX socket
- Server type: MySQL
- Server version: 5.5.31 - Source distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

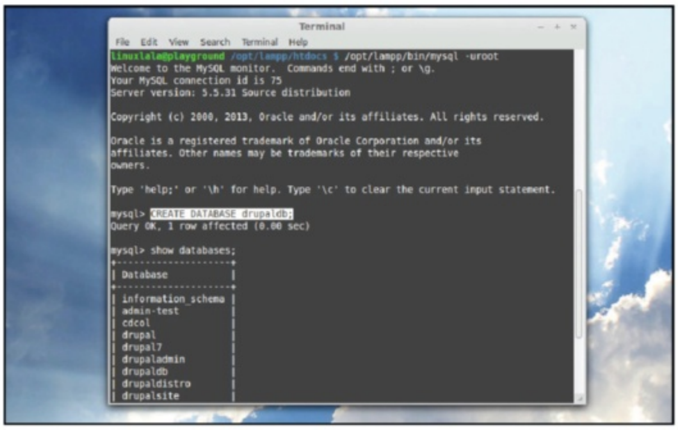
Web server
- Apache/2.4.4 (Unix) OpenSSL/1.0.1e PHP/5.4.16 mod_perl/2.0.8-dev Perl/v5.16.3
- Database client version: libmysql - mysqlnd 5.0.10 - 20111026
- PHP extension: mysqli

phpMyAdmin
- Version information: 4.0.4.1, latest stable version: 4.0.7
- Documentation

localhost / localhost | phpMyAdmin 4.0.4.1 - Mozilla Firefox
phpMyAdmin
Databases
SQL
Status
Users
Export
Import
Settings
More

4 Access PHPMyAdmin

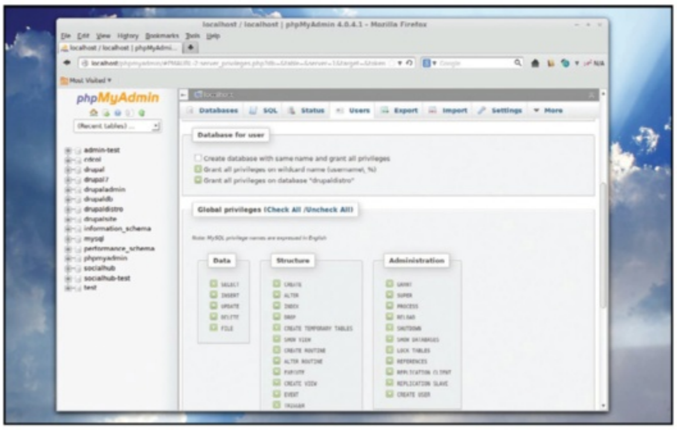
Log in to *PHPMyAdmin* as the administrator. Apart from creating users and databases, you can also use *PHPMyAdmin* to modify databases or execute SQL statements. Click the Databases button on the top-left of the main frame.



```
linuxlala@playground /opt/lampp/htdocs $ /opt/lampp/bin/mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 75
Server version: 5.5.31 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE drupaldb;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| admin-test         |
| cdcol              |
| drupal             |
| drupal7            |
| drupaladmin        |
| drupaldb           |
| drupaldistro       |
| drupalsite         |
```

5 Create the database

Fill in the name for the database and click the Create button. Existing databases are listed on this page. You will find that the newly-created database is listed along with them.



Database for user
□ Create database with same name and grant all privileges
☑ Grant all privileges on wildcard name (username_%)
☑ Grant all privileges on database "drupaldistro"

Global privileges (Check All / Uncheck All)

Note: MySQL privilege names are expressed in English

Data
☑ SELECT
☑ INSERT
☑ UPDATE
☑ DELETE
☑ FILE

Structure
☑ CREATE
☑ ALTER
☑ INDEX
☑ DROP
☑ CREATE TEMPORARY TABLES
☑ SHOW VIEW
☑ CREATE ROUTINE
☑ ALTER ROUTINE
☑ EXECUTE
☑ CREATE VIEW
☑ EVENT
☑ TRIGGER

Administration
☑ GRANT
☑ SUPER
☑ PROCESS
☑ RELOAD
☑ SHUTDOWN
☑ SHOW DATABASES
☑ LOCK TABLES
☑ REFERENCES
☑ REPLICATION CLIENT
☑ REPLICATION SLAVE
☑ CREATE USER

localhost / localhost | phpMyAdmin 4.0.4.1 - Mozilla Firefox
phpMyAdmin
Databases
SQL
Status
Users
Export
Import
Settings
More

6 Create the user

Click the Check Privileges link next to the database name. Click Add User under the New heading. Fill in the username and password and grant all privileges to the user. Click Go at the bottom of the page. LXF

An introduction: to Python

Learn the basics of the Python programming language, **Daniel Samuels** shows you the ropes.

Our expert

Daniel Samuels started learning to program at 14 and worked as a paid programmer while still at school. These days he's the lead dev at Onespacemedia, a digital creative agency based Cambridge.

In this tutorial we're going to take a look at some of the basic concepts of programming and how they're used in Python applications. You've probably heard of Python before, it's an extremely popular programming language within the technology community, and used by individuals and organisations, for large and small projects, on both the desktop and on the web.

It's popularity stems from its simple syntax which is easy to learn, a standard library which contains nearly every utility you will ever need and a supporting repository of libraries for everything else. Python is able to run on most operating systems and Linux distributions generally come with it pre-installed or make Python available through their respective package managers.

To begin writing Python open up a terminal and type **python**. If Python is installed correctly, it will load up into the interactive shell, this is a full Python environment which allows you to write and run any Python code in the same way you can with a .py file. If you didn't get a Python shell you may need to download and install it from the Python website – which is **www.python.org/download** – or from your distro's package manager.

Writing our first lines of code

Now that we have Python running, we can start writing some code! Lets start with the usual and make some text appear, type the following: **print "Hello World!"** and press Return to make it run. You should see a line of text (known as a 'string') saying Hello World! You can change this text to be almost anything you want and Python will print it to the screen, this is the basis of most Python applications. Next we're going to start storing some data and to do that we're going to use something called a 'variable', a variable is just a way of naming something so you can get it back later. To define a variable, you just need a name for it and something to store, here's an example of a variable and how we would use it:

```
fruit = 'Banana'
```

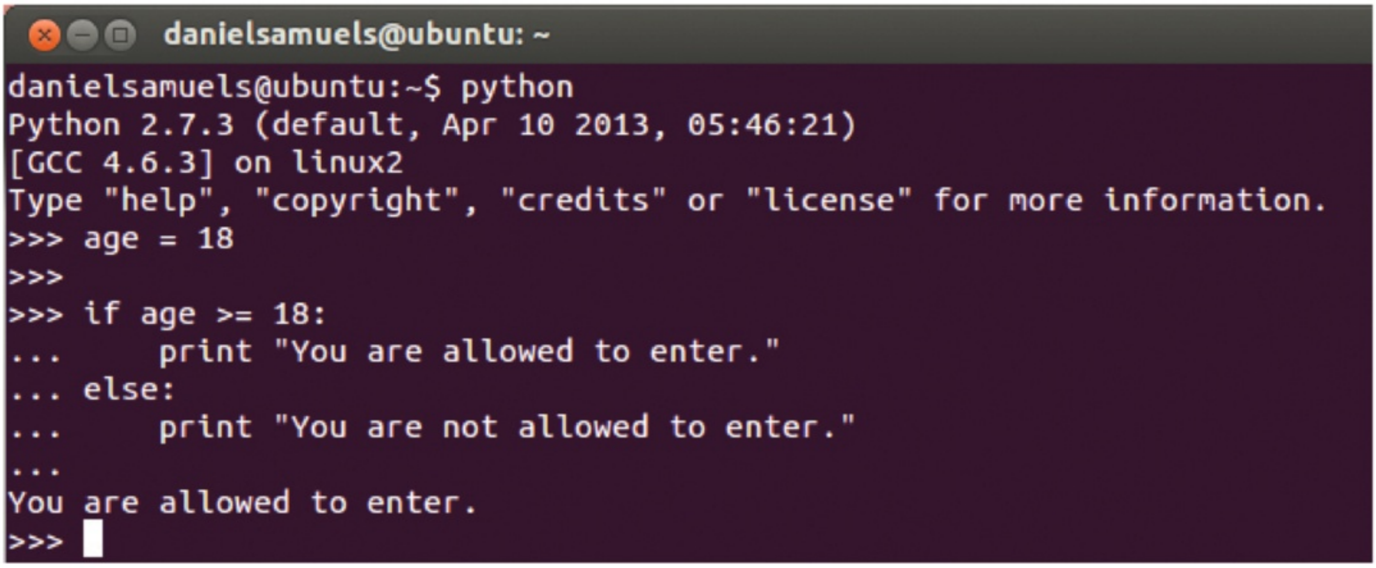
You won't see anything output from setting a variable, but we can combine it with our print from earlier to display the name of your fruit: (Note: lines prefixed with '>>' are showing how the expected output is displayed):

```
print fruit
>> Banana
```

Python allows us to perform many different actions with a string, such as finding its length, counting the number of times a letter appears, changing the case or simply taking a snippet of it. Take a look at these examples:

```
len(fruit)
>> 6
fruit.count('a')
>> 3
fruit.upper()
>> BANANA
fruit[0:3]
>> Ban
```

We can also combine the printing of strings and variables to make customised outputs, such as declaring what our favourite fruit is:



```
danielsamuels@ubuntu:~$ python
Python 2.7.3 (default, Apr 10 2013, 05:46:21)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> age = 18
>>>
>>> if age >= 18:
...     print "You are allowed to enter."
... else:
...     print "You are not allowed to enter."
...
You are allowed to enter.
>>>
```

› **This is how your Python shell should look when running the string examples.**

```
print “My favourite fruit is a “ + fruit
>> My favourite fruit is a Banana
```

With a special function named format, we can also put variables inside of our strings to allow for a more dynamic text output. This might be used to welcome a user on your application by saying ‘Hello, <name>.’ or to show the price of an item in a web shop:

```
print “I bought a {} from the shop.”.format(fruit)
>> I bought a Banana from the shop.
```

In addition to using variables for making and formatting lines of text, we can also use them to do maths, store data for later use, make decisions and a whole host of other things. Here’s a few more examples of using variables:

```
number1 = 10
number2 = 5
number1 + number2    # Add two numbers together.
>> 15
number1 * number2    # Multiply two numbers together.
>> 50
number1 ** number2    # Raise number1 to the power of
number2 (xy)
>> 100000
fruit = “Banana”
fruit * 3            # Repeat the string ‘Banana’ 3 times.
>> BananaBananaBanana
```

The last example highlights that rather than doing a multiplication, as it would if fruit was a number, Python knows to repeat ‘Banana’ three times due to it being a string.

Application logic

Logic is a major part of programming; knowing when to show a username on a webpage or a button within a desktop application is vital. The simplest way to write logic within Python, and indeed most programming languages, would be to use an if statement. We can define what we want to check and we can run different code depending on whether the check passes or not, thus allowing us to control the flow of our application. As an example, you may only want to display some content if a user is above a certain age and a simple if statement would be able to achieve that with ease.

```
age = 18
if age >= 18:
    print “You are allowed to enter.”
else:
    print “You are not allowed to enter.”
>> You are allowed to enter.
```

You may need to press Return twice for the statement to run. As you can see, we’re checking to see whether our variable age is greater than or equal to 18, if it is then they are allowed to enter, otherwise they’re not.

Using files

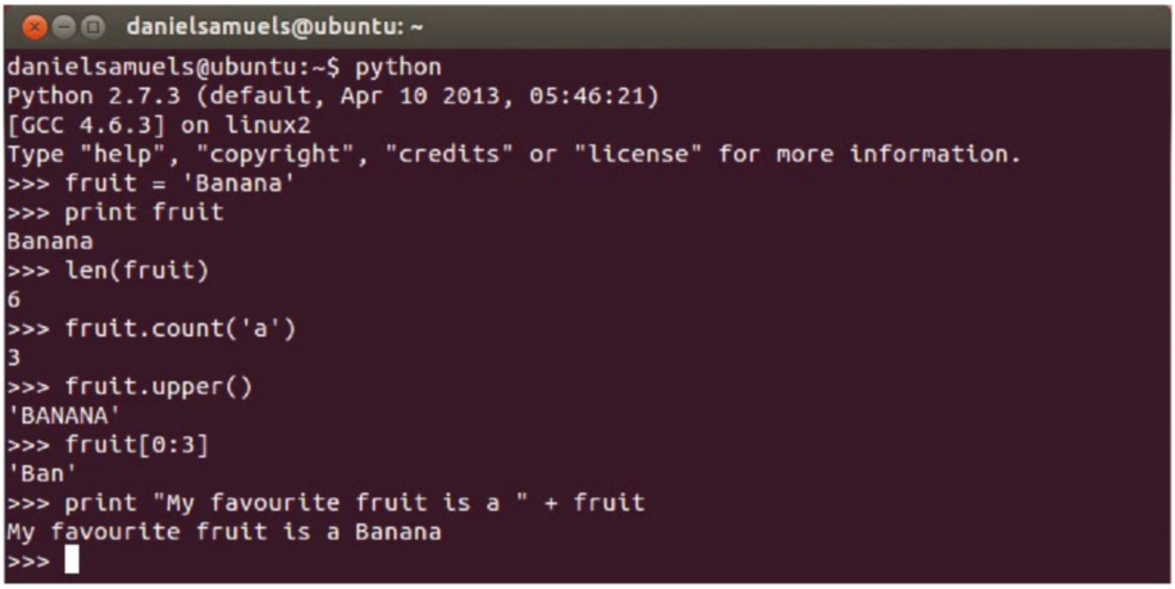
One of the issues with the Python shell is that it’s not very easy to save your work for a later date, so lets start using a .py (PY) file to save what we’re doing. Exit the Python shell by entering **exit()** and hitting Return, then open up your text editor of choice and save an empty file somewhere named **tutorial.py**. In your terminal, change into the folder where your new PY file is. We are now able to make changes to the PY file in the text editor and then simply run python **tutorial.py** in our terminal to run the code. Lets give a try:

```
tutorial.py
print “We’re now using a .py file!”
/code ends/
Then in the terminal:
/code/
python tutorial.py
>> We’re now using a .py file!
```

Go ahead and try writing some of your own variables, strings and if statements in your new PY file and see what happens when you run the file in your terminal, you can use the same code we wrote earlier. You may see a .pyc generated when you run your PY file, but don’t worry too much about it, it’s just what Python compiled your code into.

Taking your next steps

The different things we learned today aren’t specific to Python, nearly every programming language will have variables, strings, functions and methods, though they won’t necessarily be written in the same way. If you want to learn more about Python, we would recommend visiting the official website **https://wiki.python.org/moin/BeginnersGuide**. It has pages of links to other useful resources which will allow you to take your programming to the next level, we’ve barely scratched the surface. If you like learning to program in an interactive manner, we’d also recommend Codecademy (**www.codecademy.com/tracks/python**), the lessons are very easy to follow.



```
danielsamuels@ubuntu:~$ python
Python 2.7.3 (default, Apr 10 2013, 05:46:21)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> fruit = 'Banana'
>>> print fruit
Banana
>>> len(fruit)
6
>>> fruit.count('a')
3
>>> fruit.upper()
'BANANA'
>>> fruit[0:3]
'Ban'
>>> print "My favourite fruit is a " + fruit
My favourite fruit is a Banana
>>> 
```

› Running more complex code does work in the shell, though it can become hard to manage.

SAVE UP TO 50% On the perfect gift this Christmas

With over 60 titles you'll find the perfect present for that special someone

SAVE UP TO **40%**
FROM **£23.49**

SAVE UP TO **30%**
FROM **£26.99**

SAVE UP TO **40%**
FROM **£25.49**

SAVE UP TO **50%**
FROM **£12.99**

SAVE UP TO **35%**
FROM **£23.49**

SAVE UP TO **50%**
FROM **£13.99**

SAVE UP TO **50%**
FROM **£12.99**

SAVE UP TO **35%**
FROM **£22.99**

SAVE UP TO **45%**
FROM **£19.49**

SAVE UP TO **45%**
FROM **£19.49**

SAVE UP TO **47%**
FROM **£18.49**

SAVE UP TO **46%**
FROM **£20.99**

- **Save up to 50%** off the cover prices
- **Delivery included** in the price
- **Over 60 magazines** covering cars and cycling, crafts, film, music, technology, gaming and more
- **Free gift card** you can personalise to announce your gift
- Huge range of items priced **under £20**
- **PLUS!** Overseas discounts available

2 easy ways to order

myfavouritemagazines.co.uk/Z401

Or call us on **0844 848 2852**
quote **Z401**
Lines open Mon to Fri 8am – 9.30pm
and Sat 8am – 4pm

Savings compared to buying 2 year's worth of full priced issues from UK newsstand. This offer is for new print subscribers only. You will receive 13 issues in a year except Computer Music Specials, SFX Specials and Computer Arts Collection which publish 6 issues per year, and Comic Heroes which publishes 4 issues per year. Full details of the Direct Debit guarantee are available upon request. If you are dissatisfied in any way you can write to us or call us to cancel your subscription at any time and we will refund you for all unmailed issues. Prices correct at point of print and subject to change. For full terms and conditions please visit: www.myfavm.ag/magterms Offer ends: 31st January 2014

THE ALL-NEW

net

MAGAZINE IPAD EDITION

The iPad edition of **net** has been completely rebuilt from the ground up as a tablet-optimised reading experience.

The voice of web design
net
Issue 248 : December 2013 : net.creativebloq.com
PROJECT
MAKE YOUR SITE WORK FOR TOUCH
How to handle touch events in JavaScript
REINVENT!
CREATE THE PERFECT REDESIGN
Dan Mall reveals the best strategies to redesign the right way
GALLERY
Presenting this month's outstanding
PLAY-DOT-TO.COM
ALQUIMIAWRG.COM
CREATIVEMORNINGS.COM

TRY IT FOR FREE TODAY!

You'll find additional imagery, exclusive audio and video content, and we've partnered with **Tuts+ Premium** to bring you superb screencasts that tie in with the practical projects in every issue. **Don't miss it!**

TRY IT FOR FREE TODAY WITH OUR NO-OBLIGATION 30-DAY TRIAL AT
UK: netm.ag/itunesuk-248 US: netm.ag/itunesus-248

Available on the App Store

Fortran: Get started

Fortran has been around for a long time, but it's still in use, and **Juliet Kemp** finds it's easier than its reputation might suggest.

Our expert

Writer Juliet Kemp finds that there's something strangely pleasing about working in a language her Dad used to use. Though she is glad things have progressed past punch cards.

Fortran is by anyone's standards, an old language (*see the boxout Fortran Versions at the bottom of this page for more details on the history*). Despite its age, it's still used extensively in the sciences, in high-performance computing, and in supercomputers (to name a few places); and although people have been predicting its imminent demise for a good couple of decades now, it's still going strong. What Fortran is great for is numerical computation, which is why it's going to be hard to shift from its engineering, physics and maths strongholds. It's also used by some investment banks and insurance companies for complicated mathematical models.

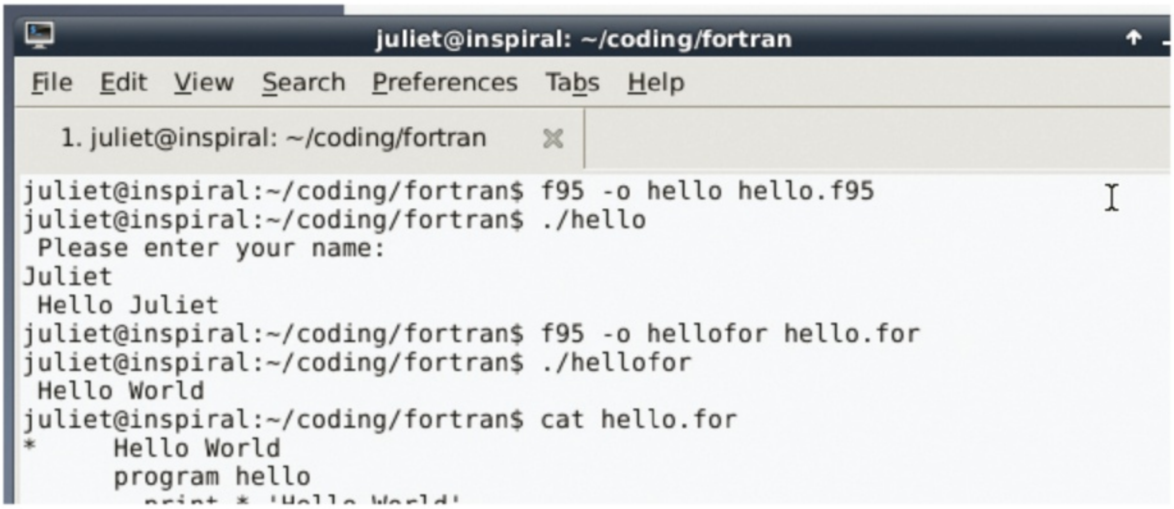
Fortran code that's 35 years old is still running and under active development in some areas. Admittedly, if you're after something for web development or GUI applications, you probably want to look elsewhere. But as with all the languages in this series, it's always interesting to have a poke around and see how you get on with it.

Getting started

As per the boxout about Fortran versions, in this tutorial we'll use Fortran 90, which means finding a Fortran 90 compiler that you like. The easiest to get hold of is probably the GNU compiler *gfortran*. Other compilers available include *G95* (free) and *NAG* (paid). Some compilers have additions to the Fortran standard, which can be useful but which lock you into using that compiler; best practice is to stick with the official standard. For this tutorial, I'll use *gfortran*, the GNU fortran compiler, which is available from the GNU website or as a package for most distros. On Debian, you can run it with either **gfortran** or **f95** (both run the same compiler); I'll use **f95** in the text.

Open up a file **hello.f95** and enter the following code:

```
! Hello World
program hello
  print *,"Hello World"
```



juliet@inspiral: ~/coding/fortran
File Edit View Search Preferences Tabs Help
1. juliet@inspiral: ~/coding/fortran

```
juliet@inspiral:~/coding/fortran$ f95 -o hello hello.f95
juliet@inspiral:~/coding/fortran$ ./hello
 Please enter your name:
Juliet
 Hello Juliet
juliet@inspiral:~/coding/fortran$ f95 -o hellofor hello.for
juliet@inspiral:~/coding/fortran$ ./hellofor
 Hello World
juliet@inspiral:~/coding/fortran$ cat hello.for
*     Hello World
      program hello
        print *,'Hello World'
```

› **f95 program which asks for your name and greets you; then the F77 version of Hello World (see boxout for more on fixed format).**

Fortran versions

Fortran (or rather, its first working compiler) appeared in 1957, as an alternative to assembly language for programming mainframes. Back then, it contained an impressive 32 statements and was, of course, stored on punchcards, one per line. It quickly grew in popularity – well, wouldn't you take it over Assembler? This meant that compilers for different types of mainframe were rapidly developed by competing manufacturers. That arguably makes it the first cross-architecture programming language. Fortran II, III, and IV appeared over the next couple of years, but Fortran 66 was the next big step, providing an 'industry-standard' version of the language.

Fortran 77, a decade later, addressed the shortcomings of 66, and Fortran 77 is perhaps the most historically important dialect. You can still get F77 compilers for modern computers, and you will still find F77 code out in the wild and doing its job.

After a very long wait, Fortran 90 was a major rewrite. However, all F77 features were retained, and any compliant F77 program should also be a compliant F90 program. Fortran 95 was a minor revision.

Another major revision has since been published, Fortran 2003 (and a minor update in 2008). However, if you want to learn Fortran today, it's usually advised that you start with 90/95, and then go on to learn about the new features added by 2003. This article will use Fortran 95.

Free format vs fixed format

Prior to Fortran90, Fortran code had to be written in fixed format. This stemmed from the days of punch cards, and had rules like this:

» Maximum line length 72 characters. To continue a line, use any character in column 6 on the next line.

» The first 6 columns must be empty unless you're using one column for a comment or continuation character.

» Comment lines have ***** or **c** in first column.

» Spaces are ignored altogether.

Our basic "hello" program would look like this:

```
c     Hello World
      program hello
        print *,'Hello World'
      end program hello
```

As of F90, you can write in free format instead. The code in this tutorial is all written in free format. There are still a few rules:

» Lines can be up to 132 characters; use **&** to continue a line. If the split is in the middle of a name, use another **&** at the start of the next line:

```
character :: name*100
name = "this is a very very long line, so you
need an ampersand here &
      &and an ampersand here as well"    !
comment
```

» Comments start with **!**

» Spaces are significant; you may not have embedded spaces in variable names or in numbers. You canr omit the space in a few terms like **END IF**, and indentation is ignored.

» You can put multiple statements on one line by separating them with **;**.

The GCC compiler assumes free format if the source filename ends in **.f95** or **.f90** and fixed format for **.for** source files. You can also specify either **-ffixed-form** or **-ffree-form** on the command line to override that default.

```
end program hello
```

Compile it with **f95 -o hello hello.f95**, then run it with **./hello** (try doing this without the **-o hello** and you'll find that the default output file is called **a.out**. This will run fine, but it's a nuisance not knowing what source file it corresponds to, and it's easy to accidentally overwrite it). You should get the traditional **Hello World** output.

These few lines show several basic language features. **!** is used for comments. Programs are started and ended with **program name** and **end program name**; the program name doesn't need to match the name of the file. Indentation is not required, but is good practice for code readability.

print prints to screen (the asterix means "use the appropriate print format for what is being output"). In F77 you had to use single quotes, but in F90 onwards double quotes are also legitimate.

Language basics

Let's try a slightly more complicated version of **hello.f95**:

```
! Hello Name
program hello
  implicit none
  character :: name*20
  print *, "Please enter your name: "
  read *,name
  print *,"Hello ",name
end program hello
```

The **implicit none** line is important. Old-school Fortran (that is, F77 and earlier) had implicit variable typing. Variable names starting with i, j, k, l, m, or n were taken to be integers (this being a language written by scientists, who automatically expect i, j, etc, to be integers), and all other names were taken to be real numbers. Strings or characters had to be explicitly declared. This will still work in F95 (for backwards compatibility), but it's a great way to introduce errors. The statement **implicit none** prevents any implicit typing from happening and forces you to declare all your variables explicitly. Here, we declare **name** to be a **character** array of length 20; that is, a 20-character long string (if you want to be able to handle longer names, increase the value).

read, fairly obviously, reads from the screen into a variable. Compile and run the whole thing, and it should ask for your name then say hello to you.

Note that Fortran's basic data types are **INTEGER**, **REAL** (6 decimal digits of precision), **DOUBLE PRECISION** (13 digits of precision), **LOGICAL** (boolean; **.TRUE.** or **.FALSE.**), **COMPLEX** (complex numbers), and **CHARACTER** (as above, an array of **CHARACTER** is a string).

ALLCAPS are often used when discussing code but aren't necessarily needed for writing it. Fortran 90 also introduced derived types, which enable you to create more complex data structures.

Quick tip

If you ever find yourself debugging someone else's Fortran code, running it through different compilers (with debug options turned on) can be really helpful way to get a large chuck of information on possible problems.

Calculations and loops

Since Fortran is heavily mathematical, let's create a mathematical program. Chances are you've encountered the Fibonacci sequence before. Each number in the Fibonacci sequence is the sum of the two numbers before it. The Fibonacci sequence shows up in all sorts of places, both mathematically and in nature. Since the algorithm is iterative (you calculate a value, then use it to calculate the next value), we'll use a loop to generate it in our code.

Here's a brief program that generates the first 20 Fibonacci numbers:

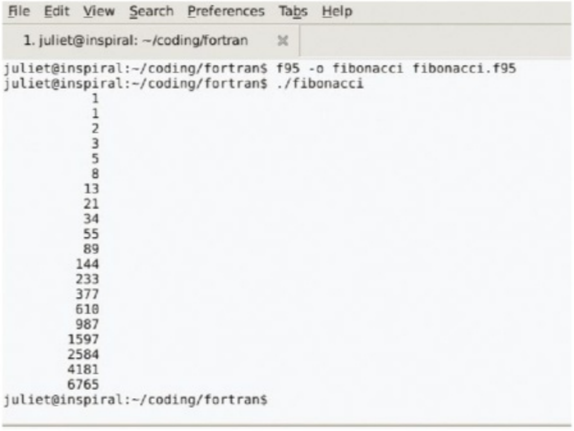
```
! Calculates the Fibonacci sequence to value 20
program fibonacci
  implicit none
  integer :: first, second, a, i
  first = 1
  second = 1
  print *, first
  print *, second

  do i = 1, 18, 1
    a = first + second
    print *, a
    first = second
    second = a
  end do
end program fibonacci
```

We start out by setting up all the variables we need; the variables to store the first two numbers in the sequence (both 1); **a**, which will be our temporary number when generating the sequence; and **i** to use as an iterator. We then print the first two numbers. The main interest is the **do** loop. In F77 and earlier you had to use labels and **GOTO** to generate a loop, »

» but as of F90 we can use a more modern syntax:

```
do iterator = start, end, increase
  ! body of loop goes here
end do
```

The iterator begins at **start**, increases by **increase**, and finishes when the iterator has value **end**. So, here **i** increases by 1 each loop from 1 to 18. Given that we already have the first 2 values of the sequence, this gives us a total of 20 values output. The body of the **do** loop is pretty straightforward; we generate and print the next value in the sequence, then we reallocate our **first** and **second** variables ready for the next loop. Note: Given that increasing by 1 is such a common use case, you could just write **do i = 1, 18** and it would increase by 1 as default. You can also adapt this program to print out exactly what it's doing during the loop, if you aren't clear on how the algorithm works.


File Edit View Search Preferences Tabs Help
1. juliet@inspiral: ~/coding/fortran

```
juliet@inspiral:~/coding/fortran$ f95 -o fibonacci fibonacci.f95
juliet@inspiral:~/coding/fortran$ ./fibonacci
           1
           1
           2
           3
           5
           8
          13
          21
          34
          55
          89
         144
         233
         377
         610
         987
        1597
        2584
        4181
        6765
juliet@inspiral:~/coding/fortran$
```

› **Generating the Fibonacci sequence.**

Calculating a function root

To show off some more complicated language features, let's try a more complicated maths problem: finding the root of a function.

Finding the root of a function means that for a function f(x), you want to find the value of x for which f(x) = 0. Here are a couple of examples:

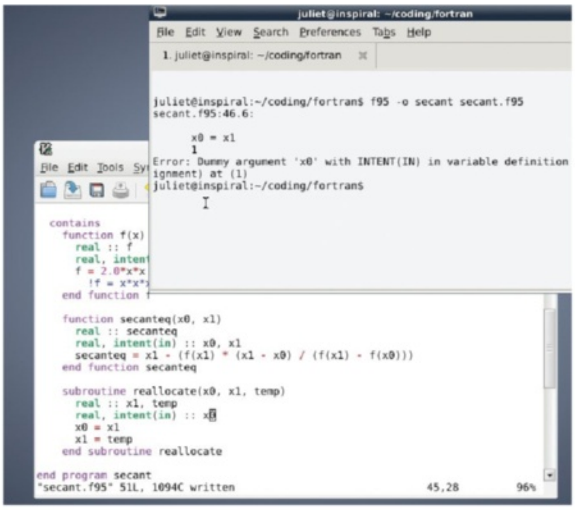
```
f(x) = x - 2 : the root of this is 2, because 2 - 2 = 0.
f(x) = 3x - 3 : the root of this is 1, because 3 * 1 - 3 = 0.
f(x) = 2x^2 - 8 : the root of this is 2, because 2 * 2 * 2 - 8 = 0.
```

This shows up in a lot of mathematical problems and situations. For all of the above functions, you could solve it mathematically (as you probably did at school). But for more complicated functions, you need a numerical computation approach to find the root.

A straightforward and very old way (even older than Fortran; 3,000 years or so) of doing this is called the secant method. I won't go into the details of how the secant method works (check Wikipedia or elsewhere if you want a full explanation), but the implementation of the algorithm below does the job!

Save the below code as **secant.f95**:

```
! Calculating the root of 2x^2-1 with secant method
program secant
  implicit none
  real, parameter :: error = 1.0e-6
  real :: x0, x1, temp
  integer :: i = 1, limit = 100
  x0 = 0
  x1 = 1.0
  do
    if (i > limit) then
      print *, "Stopping now; no convergence"
      exit
    endif
    if (f(x1) == 0) then
      print *, "Exact root is ", x1
      exit
    else if (abs(x1 - x0) < error) then
      print *, "Approx root is ", x1
      exit
    end if
```


juliet@inspiral: ~/coding/fortran
File Edit View Search Preferences Tabs Help
1. juliet@inspiral: ~/coding/fortran
File Edit Tools Sy

```
juliet@inspiral:~/coding/fortran$ f95 -o secant secant.f95
secant.f95:46.6:

     x0 = x1
     1
Error: Dummy argument 'x0' with INTENT(IN) in variable definition
ignment) at (1)
juliet@inspiral:~/coding/fortran$
```

```
contains
  function f(x)
    real :: f
    real, intent
    f = 2.0*x*x
      !f = x*x*
  end function f

  function secanteq(x0, x1)
    real :: secanteq
    real, intent(in) :: x0, x1
    secanteq = x1 - (f(x1) * (x1 - x0) / (f(x1) - f(x0)))
  end function secanteq

  subroutine reallocate(x0, x1, temp)
    real :: x1, temp
    real, intent(in) :: x0
    x0 = x1
    x1 = temp
  end subroutine reallocate

end program secant
"secant.f95" 51L, 1094C written                    45,28        96%
```

› **Here, I've used the intent(in) keyword for a variable that the subroutine wants to change (shown in the editor. The compiler has thrown an error.**

Debugging

Unfortunately, Fortran compiler messages aren't always entirely helpful. Always check that you have **implicit none** set if you're getting weird messages. You can also add **print** statements in various places to check what's happening.

After that, you can try some of the GFortran debugger options:

» **-O0**. This directs the compiler to make no optimisations. Optimising can do odd things to local variables, which can make debugging difficult. The downside is that it makes your code run very slowly. **-Og** in newer GFortran versions (post 4.8) is another option, which applies only optimisations that won't affect debugging.

» **-Wall** enables 'all' warnings; **-Wextra** enables some extra ones not included in 'all'.

» **-pedantic** warns about language features supported by gfortran, and are not part of the official standard.

» **-g** allows you to run your code through **gdb**, the GNU interactive debugger. Use this with **gdb programname**; you can then type **break main** to set a breakpoint just before the first executable statement, **run** to run to the breakpoint, and **step** to step through statements one by one.

There's plenty of online documentation for GDB, although better Fortran debuggers are also available.

```
    temp = secanteq(x0, x1)
    call reallocate(x0, x1, temp)
    i = i+1
    !uncomment the line below to see the loop in action
    !print *, "x0 is ", x0, "x1 is ", x1
  end do
  contains
    function f(x)
      real :: f
      real, intent(in) :: x
      f = 2.0*x*x - 1.0
    end function f

    function secanteq(x0, x1)
      real :: secanteq
      real, intent(in) :: x0, x1
      secanteq = x1 - (f(x1) * (x1 - x0) / (f(x1) - f(x0)))
    end function secanteq

    subroutine reallocate(x0, x1, temp)
      real :: x0, x1, temp
      x0 = x1
      x1 = temp
    end subroutine reallocate

end program secant
```

First of all, let's look at the overall structure of the program. Inside the program, as defined by the **program ... end program** lines, there is a setup section, a **do** loop, and then a **contains** section.

The **contains** section allows you to create internal subprograms (functions or subroutines) within your program. You can also create external subprograms that appear after the **end program** line. There are two main differences between a function and a subroutine:

1. A function returns a value; a subroutine does not.
2. A function must have at least one argument; a subroutine can have none.

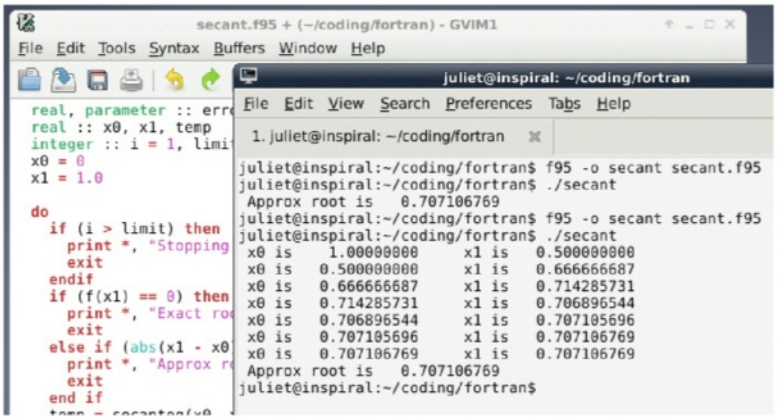
We'll look at what the functions and subroutine are doing in a moment. For now, just notice that they are defined with the keywords **function ... end function**, and **subroutine ... end subroutine**.

Let's look at what's happening in the program, then. The first few lines set up assorted variables. The only new part here is the **parameter** keyword. This is used to define a constant in Fortran. If you declare a parameter, you're asserting that the value of that parameter won't change during the program (and you'll get a compiler error if you do try to change it). We also set up **x0** and **x1**, which are our iterative values (in the same way as you saw in the Fibonacci program). The idea is that over a number of iterations, these values get closer and closer to the actual root of the equation.

Most of the program logic, again, is in the **do** loop. This time, instead of using an iterator, we have an infinite loop. If you use one of these, you want to make sure that there is some stopping condition. In this instance, we test **i** (being used as an informal iterator) against **limit** and exit the loop. The **exit** keyword jumps you out of the loop that it is in. Each run of the loop looks at our current two x-values, and either exits, or recalculates them.

After checking that the 100-round limit hasn't been reached yet, we test our most recently calculated x value, **x1**, to check if it returns 0 when plugged into the function. If it does, then we have found a root and can exit. We also check for "nearly there"; if the two most recently calculated x values, **x0** and **x1**, have a very small difference between them, then we have an approximate root and can also stop. The variable **error** sets the accuracy; here, it tests for a difference of less than 0.000001 between the values.

If neither of these are true, we're still looking for a root. We therefore calculate the secant equation for **x0** and **x1**, and save it in **temp**. The subroutine **swap** then sets the old **x1** as the new **x0**, and **temp** as the new **x1**; and back we go to the top of the loop.


secant.f95 + (~/coding/fortran) - GVIM1
File
Edit
Tools
Syntax
Buffers
Window
Help
juliet@inspiral: ~/coding/fortran
File
Edit
View
Search
Preferences
Tabs
Help
1. juliet@inspiral: ~/coding/fortran

```
real, parameter :: err
real :: x0, x1, temp
integer :: i = 1, limi
x0 = 0
x1 = 1.0

do
  if (i > limit) then
    print *, "Stopping
    exit
  endif
  if (f(x1) == 0) then
    print *, "Exact ro
    exit
  else if (abs(x1 - x0
    print *, "Approx r
    exit
  end if
```

```
juliet@inspiral:~/coding/fortran$ f95 -o secant secant.f95
juliet@inspiral:~/coding/fortran$ ./secant
 Approx root is   0.707106769
juliet@inspiral:~/coding/fortran$ f95 -o secant secant.f95
juliet@inspiral:~/coding/fortran$ ./secant
 x0 is     1.00000000      x1 is   0.500000000
 x0 is   0.500000000      x1 is   0.666666687
 x0 is   0.666666687      x1 is   0.714285731
 x0 is   0.714285731      x1 is   0.706896544
 x0 is   0.706896544      x1 is   0.707105696
 x0 is   0.707105696      x1 is   0.707106769
 x0 is   0.707106769      x1 is   0.707106769
 Approx root is   0.707106769
juliet@inspiral:~/coding/fortran$
```

› **Finding the root of this equation. You can also uncomment the print line in the loop if you wish to watch the values converging, as they are here.**

Subprograms

We have two functions in the **contains** section. One is **f(x)** – the function that we want to find the root of. The other is **secanteq(x0, x1)**, which applies the secant method equation to our two working values. Within the function, the function name is given a type and treated as a variable. This is the value that will be returned from the function, and you must explicitly declare its type, as here. You must also specify the type of the arguments. If you try to pass in a variable which does not match the argument type, you'll get a compile error.

The other new thing here is the **intent(in)** keyword. This helps to avoid accidentally editing or passing back values from a function. These values are now defined as input-only, so if the function tries to change them, a compile error will be thrown.

In the subroutine, too, the arguments must have their types defined. However, unlike a function, a subroutine doesn't return a value. Instead, it alters any values within itself (if you are passing in any values that you don't want to change, you can use **intent(in)** again). We are shuffling three values around. We discard the old **x0**, the old **x1** becomes **x0**, and the new temporary variable becomes **x1** (you could do this inline if you prefer; here I've broken it out partly to demonstrate a subroutine). Having put all that together, you can now compile it and run it, and see if you get an answer.

That should be enough to introduce you to the way Fortran works. If you're used to more modern languages, it can feel a bit clumsy. However, it is extremely well suited to certain sorts of numerical calculations; and there are huge quantities of Fortran code freely available to solve many mathematical and scientific problems. If you do find yourself in need of some Fortran code, check out the many online libraries before you roll your own. Fortran 95 allows you to use modules and other structures that make larger programs easier, though these are outside the scope of this tutorial.

Finally, a shout-out to the *Fortran Colouring Book* by Roger Kaufman, which is possibly the most entertaining computer language manual available (it's hand-written, with lots of Dr Seuss-style cartoons). Sadly, since it was written in 1978 (as an MIT textbook) it covers only F77, not F95. LXF

Answers

Got a question about open source? Whatever your level, email it to **lxf.answers@futurenet.co.uk** for a solution.

This month we answer questions on:

1 Checking your hard drive with SMART
2 Code queries
3 Redirecting command output
4 Raspberry Pi photography
5 Partitioning problems
★ Solving driver issues

1 Get SMART

Q Please can you tell me how I can check whether my computer hard disk is OK or if it is failing? I have Ubuntu 13.04 installed on my system.
Narendra
From the forums

A If you have the slightest suspicion that your hard drive is faulty or about to fail, you must back up immediately. Once your data is in a safe place, you can check out your drive, which is generally done using S.M.A.R.T (which we'll call SMART from here on for convenience). While this looks like the name of a criminal organisation from a sixties spy spoof, it stands for Self-Monitoring, Analysis and Reporting Technology.

It is a way for hard drives to monitor themselves and provide information to the OS in advance of any problems surfacing. There are a couple of conditions to using SMART. First, your drive must support it – all modern drives do. Second, it must be enabled in your computer's BIOS – so reboot into your BIOS settings and check.

The **martmontools** package will also need installing. This has two programs; **smartd** is a daemon that runs in the background monitoring your drives and alerts you of any problems. **Smartctl** is the user program for checking drives. First, make sure that SMART is available for your drive by running:

```
smartctl -i /dev/sda
```

This will print various information about your drive, and should end with:

```
SMART support is: Available - device has
SMART capability.
SMART support is: Enabled
```

If it reports unavailable, you cannot use SMART to check your drive. If it reports disabled, you didn't turn it on in the BIOS. For a quick look at the state of the drive, run:

```
sudo smartctl --health /dev/sda
```

There are a number of self-tests the drive can be instructed to perform. The short self-test can be used while the system is running and usually takes a few minutes to complete:

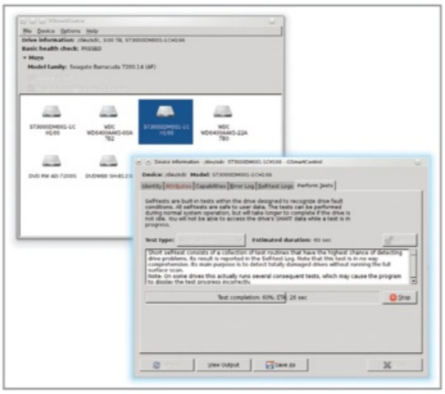
```
sudo smartctl --test=short /dev/sda
```

The terminal prompt returns immediately because the test is running in the background. You can see the current status of running tests and the results of completed tests with:

```
sudo smartctl --all /dev/sda
```

If you want to be really certain, you can also run the long test, which will be much more thorough. There is also a **--captive** option that forces tests to be run in the foreground, but this must not be used while there are any mounted partitions on the drive – it should be used only from a live CD.

The **smartd** program runs in the background, routinely checking your drives and notifying you by email if there is an error. You will need to edit **/etc/smartd.conf** to set the tests to check and the email address to use. The file contains comprehensive comments and examples. There's also a graphical alternative to **smartctl**, called **GSmartControl**, which covers the common actions of **smartctl** in a GUI.



Drive information: /dev/sdc, 3.00 TiB, ST3000DM001-1CH166
Basic health check: PASSED
Model family: Seagate Barracuda 7200.14 (AF)

Device Information - /dev/sdc: ST3000DM001-1CH166 - GSmartControl
Device: /dev/sdc Model: ST3000DM001-1CH166

Selftests are built-in tests within the drive designed to recognize drive fault conditions. All selftests are safe to user data. The tests can be performed during normal system operation, but will take longer to complete if the drive is not idle. You will not be able to access the drive's SMART data while a test is in progress.

Test type: Estimated duration: 60 sec

Short self-test consists of a collection of test routines that have the highest chance of detecting drive problems. Its result is reported in the Selftest Log. Note that this test is in no way comprehensive. Its main purpose is to detect totally damaged drives without running the full surface scan.
Note: On some drives this actually runs several consequent tests, which may cause the program to display the test progress incorrectly.

Test completion: 60%, ETA: 26 sec

› ***GSmartControl* provides a friendly UI for the SMART disk self-test tools.**

2 What about the dollar?

Q I have been googling for some help on generating a large binary file for use in testing a program I'm writing and came across this code:

```
dd if=/dev/zero of=10g.img bs=1000 count=0
```

Enter our competition

Win!

if ($letter == winner) get $books

Get your questions answered and exploit our generosity.

***Linux Format* is the biggest and best magazine about Linux and free software.**

We did a rough word count of LXF173 and it contained 67,659 words. About as much as *Heart of Darkness* and *Animal Farm* combined, but with considerably more Linux, coding and free software!

That's many more than our competitors, and that's how we know that we're the biggest. As for the best, well… that's a subjective claim, but it's one we're happy to stand by.

Because we're nice chaps as well as Linux gurus, and because we've moved to a new office that's got very little storage space, we're giving one lucky reader each issue the chance to win some of the books that get sent to us for review. They're mostly reference books that will give your shelves an air of computational gravitas. For a chance to win, email your question to **lxf.answers@futurenet.co.uk**, or post it to **www.linuxformat.com/forums** to seek help from our lively community of readers.

› See page 94 for our star question.

Terminals and superusers

We often give a solution as commands to type in a terminal. While it is usually possible to do the same with a distro's graphical tools, the differences between these mean that such solutions are very specific. The terminal commands are more flexible and, most importantly, can be used with all distributions.

System configuration commands often have to be run as the superuser, often called root. There are two main ways of doing this, depending on your distro. Many, especially Ubuntu and its derivatives, prefix the command with *sudo*, which asks for the user password and sets up root privileges for the duration of the command only. Other distros use *su*, which requires the root password and gives full root access until you type *logout*. If your distro uses *su*, run this once and then run any given commands without the preceding *sudo*.

```
seek=$[1000*1000*10]
```

I know what the dd command does, but what is the $ in the seek parameter? Also, I ran this and it produced zilch. How can the count=0?

PK Fox

A The dollar sign with brackets substitutes the result of the calculation contained in there. Try:

```
echo $[1000*1000*10]
```

although it's more common to use double parentheses for this:

```
echo $(( 1000*1000*10 ))
```

which is why you couldn't find it in the *Bash* man page. You can use variables in here too:

```
X=10
echo $(( X * 1024 * 1024 ))
```

So all it is doing is using 10G in a more readable fashion. Using the **dd** command in this way produces a sparse file, it is more common to see it used with **count=1**, but **0** works. The **seek** option tells **dd** to start that far into the file, so all the preceding space has to be created. A sparse file uses only the space needed by the data it holds, so while

```
ls -l 10g.img
```

will show a large file.

```
du - h 10g.img
```

shows the disk space it actually occupies. This is often used when creating a disk image for a virtual machine. The size shown by **ls** will remain constant, while the space it uses increases as you use it. This provides a faster virtual disk than allowing the disk file to grow as it is used, which can often lead to substantial fragmentation.

I'm not sure how it will suit your needs, as the file contains no data until you add it, but if you need to create a container file to be filled later, which is what you do with VM disk images, this is a good approach.

3 Terminal recording

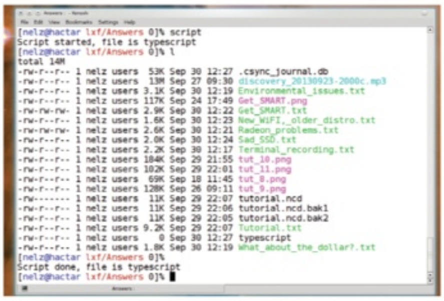
Q I have been playing with nmap and it's taking a while. If I run a command in the terminal and get output from it, is it possible to get that output piped somewhere after the command has been run, or do I have to rerun the command with the pipe?

Fat_Tuesday

From the forums

A You are really asking how to shut the stable door after the horse has bolted, but there you do have some options, depending on the type of terminal you are using. KDE's *Konsole*, which is a terminal worth considering even if you don't use KDE, has a Save Output As option in its File menu. This saves the entire scroll buffer to a file. *Gnome-terminal* doesn't currently have this feature. If you are not in X and are using a virtual console, the **setterm** command may help:

```
setterm -dump -file output.txt
```



```
[nelz@hactar lxf/Answers 0]% script
Script started, file is typescript
[nelz@hactar lxf/Answers 0]% l
total 14M
-rw-r--r-- 1 nelz users  53K Sep 30 12:27 .csync_journal.db
-rw-r--r-- 1 nelz users  13M Sep 27 09:30 discovery_20130923-2000c.mp3
-rw-r--r-- 1 nelz users 3.1K Sep 30 12:19 Environmental_issues.txt
-rw-r--r-- 1 nelz users 117K Sep 24 17:49 Get_SMART.png
-rw-rw-rw- 1 nelz users 2.9K Sep 30 12:22 Get_SMART.txt
-rw-r--r-- 1 nelz users 1.6K Sep 30 12:23 New_WiFI,_older_distro.txt
-rw-rw-rw- 1 nelz users 2.6K Sep 30 12:21 Radeon_problems.txt
-rw-r--r-- 1 nelz users 2.0K Sep 30 12:24 Sad_SSD.txt
-rw-r--r-- 1 nelz users 2.2K Sep 30 12:17 Terminal_recording.txt
-rw-r--r-- 1 nelz users 184K Sep 29 21:55 tut_10.png
-rw-r--r-- 1 nelz users 102K Sep 29 22:01 tut_11.png
-rw-r--r-- 1 nelz users  69K Sep 18 11:45 tut_8.png
-rw-r--r-- 1 nelz users 128K Sep 26 09:11 tut_9.png
-rw------- 1 nelz users  11K Sep 29 22:07 tutorial.ncd
-rw-r--r-- 1 nelz users  11K Sep 29 22:06 tutorial.ncd.bak1
-rw-r--r-- 1 nelz users  11K Sep 29 22:05 tutorial.ncd.bak2
-rw-r--r-- 1 nelz users 9.2K Sep 29 22:07 Tutorial.txt
-rw-r--r-- 1 nelz users    0 Sep 30 12:27 typescript
-rw-r--r-- 1 nelz users 1.8K Sep 30 12:19 What_about_the_dollar?.txt
[nelz@hactar lxf/Answers 0]%
Script done, file is typescript
[nelz@hactar lxf/Answers 0]%
```

Run script and all terminal input and output will be recorded until you stop it.

This, or the corresponding **append** command, writes the console output to a file, but only the visible part of the buffer, so it's not massively useful.

Your best option is to get into the habit of redirecting command output to a file when you need it. If you need to see it as well, you can use **tee**, which sends its input to both standard output and a file:

```
somecommand | tee somefile.log
```

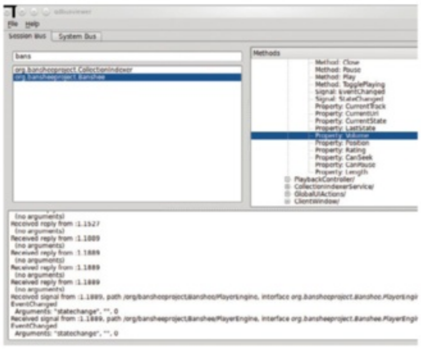
There is another option (isn't there always with Linux?), and that is called **script**. You run this once in a terminal and all output from then on is saved to a file. It works by launching another shell within the one you are using, so it has control over the input and output streams, and saves everything until you log out of that shell with Ctrl+D, at which point you will be back in your original shell with no apparent difference, except there will be a file called **typescript** containing all output from that session. You wouldn't notice that you were in a different shell, your environment is inherited by it, and it is hardly worth mentioning – except for one thing. Because you are now running in a "sub-shell", you have a different history, so when you exit **script**, the up arrow key will recall the commands you ran before running »

A quick reference to...

D-Bus

D-Bus, or Desktop Bus to use its full name, is a low-level inter-process communication system used by Linux and other POSIX operating systems. It is a means for programs to send messages to one another, either to ask for information or send commands or requests. There were previously at least two incompatible systems in use: KDE's DCOP, and Bonobo used by Gnome. D-Bus is influenced heavily by the design of DCOP, but is desktop agnostic. In fact, despite the name, it is not reliant on any desktop at all. There are generally two buses in use on a desktop computer, a system bus used by root programs such as daemons, hardware detection software and the like, and a session bus which is used by the desktop.

See the question on Silent Banshee in **LXF156** for an example of how D-Bus can be used, but there is far more to it and many programs are using it behind the scenes, from hardware detection to system notifications. The commands used to control it from the shell may look unwieldy, what with the long service names and object paths, but that is necessary if all programs are able to provide a D-Bus interface with none of them clashing. Using *qdbusviewer* to browse the tree, run commands and view the status is the easiest way to learn. If you want to access a program's D-Bus interface through a script, *qdbus* is the easier to use of the options. There is also *dbus-send*, but that works at a lower level and will require some man page reading before you can use it to do what you want.



The secrets of D-Bus revealed!

script, not those while it was in use. If you need those commands, they are saved in the **typescript** file along with their output.

4 Environmental issues

Q I have a Raspberry Pi with a really useful Adafruit LCD display fitted, which is set up to run Arch. I've written a Python script to run at boot, which uses *gphoto2* to control a Nikon digital camera in order to take timelapse photographs.

The script allows the user to set the number of images to take, and the time between each shot, using the LCD plate.

The script uses the command gphoto2 -set-config-value to instruct the camera to save the images on the SD card in the camera. If I run the script under Python (which has to be done with sudo to run the Adafruit modules), it runs correctly and saves them on the camera. If I run it through systemd with a service script linking to the exact same file, either at boot or starting it with sudo systemctl start gphoto.service, it executes the command to set the config value, but then appears to ignore it.

The script executes without error, but the images are not saved. I have added logging to the script, which shows the command being executed.

I have tried this on a laptop and the same thing happens, so I suspect this is an issue with systemd (or my use of it) rather than the Pi.

Phil Myott

A This is not an issue with **systemd**, but with the environment in which the script is being run. I see the exact same behaviour when running a script containing the commands you used from cron. There is a difference between running a script or program from a daemon such as **systemd** and from a shell. When you open a shell session, a specific environment is set up – that environment is not present when running from a daemon, even though it is running as the same user. The issue here is not that you are running the script from **systemd** but that you are not running it from a shell session.

The solution is simple, even if it seems a little convoluted. If your script were a shell script, you could load the profile as the first line of the script with one of:

```
source /root/.bash_profile
source /root/.profile
```

depending on the shell you are using to run the script. The **source** command executes the contents of the profile file within the current shell session. Normally, executing a script within a shell runs it in a sub-shell, leaving the current shell unaffected – you can test this by running a script that contains only a **cd /some/path** command and looking at the current directory in the shell after running it. As you are using a Python script, you need to launch that from within a shell session containing the correct environment, which you can do with a wrapper script:

```
#!/bin/bash
source /root/.bash_profile
/home/pi/timelapse.py
```

This sets up the correct environment and then runs the Python script, which inherits that environment. You may also be able to do this in the service file without a separate script, with something like:

```
ExecStart=bash --login -c “python /home/pi/timelapse.py”
```

I haven't tested this approach, and I would lean towards the wrapper script approach, as it keeps things simpler and easier to debug, at the expense of one more file.

5 Sad SSD

Q I have an Asus 701 Eee PC with 4GB, which I've been using while we travel, but now I'm having trouble. I tried to install another OS (which I do occasionally), but everything went west. I was able to recover my files, but now I don't have any system at all on it. I made a USB with Gparted and did an fdisk -l and checked the drive: seems OK. Then I did a Gparted, and this is where I got lost, I couldn't understand the bits of setting up partitions. When I finally think I've got it and try to install an OS, I get a message "disk error, press any key to restart", but nothing happens.

Ken Murphy

A **Fdisk** doesn't really check the drive, it only reads the partition table in the first 512 bytes of the device. Distro installers no longer need you to worry about partitioning, they will take care of it for you. There is usually an option to "Use entire disk",

Star Question **Winner!** This month's winner is Bryan Mitchell. Get in touch with us to claim your glittering prize!

★ Radeon problems

Q I responded to an update for Ubuntu 13.04. After the requested reboot, my computer became in effect unusable, because the background filled the screen entirely, and therefore I could not see any icons nor the top menu. As I have had screen problems with all Linux operating systems on this computer, I tried restarting by first substituting "nomodeset" for "quiet splash", as this has worked before. After doing this, I would then load my commercial graphics card driver. This time, it did not work. I have had to wipe the hard drive and re-install from a backup with the aid of Clonezilla. I have an older computer running 13.04 with no problems at all!

The problem computer has an i3-3220 processor, $GG TAM and a Radeon HD6450 graphics card. Can you tell me if this is a hardware problem, please?

Bryan Mitchell

A It is more likely to be a driver problem. You mention loading the commercial driver, but Ubuntu 13.04 includes the open source Radeon driver, which supports your card. If you have both drivers installed, there is plenty of potential for things to go wrong. Using the open source driver is the best option, it supports 3D on your card and has better support from the Ubuntu community. To do this, you need to purge the fglrx commercial driver from your system and re-install a few other packages. Because you are removing the graphics driver and re-installing X software, you should do this without the desktop running. To switch to a console and close the desktop, first log out of the desktop in the usual way, to make sure anything you are using is shut down cleanly. When you are back at the graphical login window, press Ctrl+Alt+F1 to get to a virtual console and log in there. Then shut down the desktop with:

```
sudo service lightdm stop
```

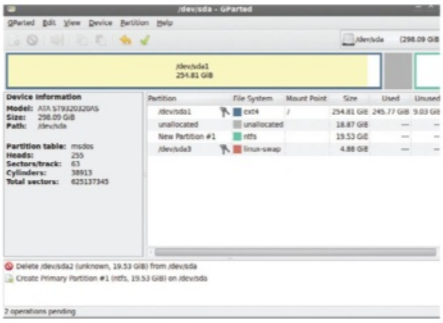
Now clear out all traces of the fglrx driver with these commands:

```
sudo apt-get remove --purge xorg-driver-fglrx fglrx*
sudo apt-get install --reinstall libgl1-mesa-glx libgl1-mesa-dri xserver-xorg-video-ati xserver-xorg-video-radeon xserver-xorg-core
sudo dpkg-reconfigure xserver-xorg
```

This clears out all the proprietary driver files and makes sure the correct open source drivers are installed. Now reboot, and then run the following command:

```
dmesg | egrep ‘drm|radeon’
```

This should show the Radeon driver (that's the name of the open source driver) being loaded. By sticking to the open source driver, you can avoid such problems with upgrades because the distro developers are able to make sure that all components of the distro work together, a control they do not have with proprietary software.



/dev/sda - GParted

/dev/sda (298.09 GiB

/dev/sda1
254.81 GiB

Device information
Model: ATA ST9320320AS
Size: 298.09 GiB
Path: /dev/sda

Partition table: msdos
Heads: 255
Sectors/track: 63
Cylinders: 38913
Total sectors: 625137345

Partition	File System	Mount Point	Size	Used	Unused
/dev/sda1	ext4	/	254.81 GiB	245.77 GiB	9.03 GiB
unallocated	unallocated		18.87 GiB	---	--
New Partition #1	ntfs		19.53 GiB	---	--
/dev/sda3	linux-swap		4.88 GiB	---	--

Delete /dev/sda2 (unknown, 19.53 GiB) from /dev/sda
Create Primary Partition #1 (ntfs, 19.53 GiB) on /dev/sda

2 operations pending

› ***GParted*** **is good for partitioning disks, but most distro installers can take care of this.**

or words to that effect. Pick this option and the installer will delete any existing partitions and create what it needs. If you prefer, you can use *Gparted* to remove the existing partitions, but this is not necessary. With only 4GB to play with, there is no space for fancy partitioning arrangements, you just need a swap partition, and the rest is for the root filesystem. Once again, any distro installer will be able to handle this for you.

However, the Eee PC was a cheap and cheerful device, and your 701 must be many years old by now. I had a later model, the 901, and its SSD drive failed some years ago. The Eee PC does not use the sophisticated SSD drives available nowadays, with wear levelling to improve their lifespan, so I wouldn't be surprised to find that the SSD had failed. This component is surface-mounted, replacement is not economical, but you can get very low-profile USB memory sticks nowadays. I have a 16GB drive that protrudes only 7mm and cost less than £10. You could install a distro onto one of these to extend the life, and capacity, of your netbook.

6 New Wi-Fi, older distro

Q I have a Toshiba Satellite laptop, running Ubuntu 12.04. I have a problem in that my Wi-Fi driver, rtl8723ae, is not supported. Can you inform me the best way to install the driver, or is it supported in the later 13.04?
Ray Hughes

A This is a relatively new chipset, and it takes time for drivers to filter through to the distros. First, the driver details have to be obtained, then they have to be added to the latest testing kernel, then they filter through to the stable kernel releases, and then you have to wait for your distro to release a version with at least that kernel. The first stable kernel branch to include the rtl8723ae driver was 3.8, and this is the one used by Ubuntu 13.04. It is possible to download the driver direct from Realtek, compile it yourself and hope it works (the separate driver is rather fussy about kernel versions), or you could try NDISwrapper to use the Windows driver. You could also try installing a 3.8 series kernel (or later) on Ubuntu 12.04, but by far the simplest solution is to upgrade to 13.04. You don't need to re-install to do this, you can use the **dist-upgrade** feature to do it, either through the software centre or by opening a terminal and running:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

By the time you read this, Ubuntu 13.10 will be available, or nearly, so you may want to jump straight to the latest version. Once a driver is in the kernel, it is rarely removed, so your Wi-Fi card should be supported by all kernel releases, and therefore just about all distros, for the foreseeable future. LXF

Help us to help you

We receive several questions each month that we are unable to answer, because they give insufficient detail about the problem. In order to give the best answers to your questions, we need to know as much as possible.

If you get an error message, please tell us the exact message and precisely what you did to invoke it. If you have a hardware problem, let us know about the hardware. If Linux is already running, you can use the *Hardinfo* program (**http://hardinfo.berlios.de**) that gives a full report on your hardware and system as an HTML file you can send us.

Alternatively, the output from *lshw* is just as useful (**http://ezix.org/project/wiki/HardwareLiSter**). One or both of these should be in your distro's repositories. If you are unwilling, or unable, to install these, run the following commands in a root terminal and attach the **system.txt** file to your email. This will still be a great help in diagnosing your problem.

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

Frequently asked questions…

KVM

› ***KVM*, isn't that a box for sharing the same keyboard and monitor between two computers?**
Yes, it is, but that's a different *KVM*; this one is a Kernel-based Virtual Machine.

› **Urk! I think the other one is easier to understand. What does this do?**
It is a way of running virtual machines more efficiently (or faster) using the virtualisation extensions built in to recent Intel and AMD CPUs.

› **So it's hardware and not a function of Linux?**
To an extent. The extensions are built into the hardware, but the kernel has the software that takes advantage of them.
This used to be done with a separate package of modules, but nowadays, it is all build in to the kernel.

› **How do I know if my CPU and kernel support *KVM*?**
Run **cat /proc/cpuinfo** in a terminal. If the flags section contains either **svm** (for AMD) or **vmx** (for Intel), you have a suitable CPU. If you are using a reasonably current distro release, you almost certainly have *KVM*. To be sure, run

```
sudo modprobe -l kvm\*
```

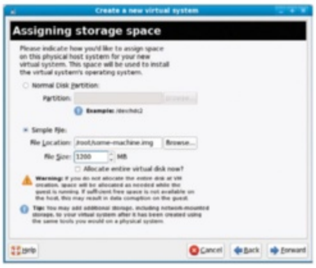
and if you get any output, *KVM* is enabled in your kernel.

› **So, I've got *KVM* in my CPU and kernel, what do I have to do to get the extra performance for virtual machines?**
Most of the time, nothing. *VMware* and *VirtualBox* can use the *KVM* features automatically, although you may need to make sure that the kvm-intel or kvm-amd modules are loaded (add them to your distro's list of modules to be loaded at boot time).

› **Is there anything else that uses it?**
There is a version of *Qemu* that is designed specifically to work with *KVM*. While the normal *Qemu* emulates the processor as well as the rest of the machine, which can be unbearably slow, the *KVM* version is very fast. The bare program doesn't have the bells and whistles of *VMware* or *VirtualBox*, but it is open source and truly free.

› **You make it sound a bit unfriendly. I've got used to *VirtualBox* now, is it that bad?**
Qemu is command-line based (running it, not the virtual machines themselves), but there is a handy graphical manager for *KVM* virtual machines called – surprise, surprise - *Virtual Machine Manager* (it's probably in your package manager as **virt-manager**).

› **This all sounds very good, are there any problems?**
KVM has been around for a while now, so it is generally stable and reliable. However, trying to run two different virtualisation systems at the same time, say *VirtualBox* and *Qemu*, can cause lockups as both try to use the CPU extensions at the same time.



Create a new virtual system

Assigning storage space

Please indicate how you'd like to assign space on this physical host system for your new virtual system. This space will be used to install the virtual system's operating system.

○ Normal Disk Partition:
Partition:
Example: /dev/hdc2

● Simple File:
File Location: /root/some-machine.img
File Size: 1200 MB
□ Allocate entire virtual disk now?

Warning: If you do not allocate the entire disk at VM creation, space will be allocated as needed while the guest is running. If sufficient free space is not available on the host, this may result in data corruption on the guest.

Tip: You may add additional storage, including network-mounted storage, to your virtual system after it has been created using the same tools you would on a physical system.

› **The *Virtual Machine Manager* can create and manage VMs for *KVM*-based virtualisation systems.**

On the disc

Distros, apps, games, podcasts, miscellany and more...

The best of the internet, crammed into a massive 4GB of quality DVD.

Distros

Choice is good, that's one of the mantras of the Linux world. So change must be good too, right? After all, that's where choice comes from. But change for the sake of change? What's the point?

We have been experimenting with using *Grub* to boot the LXFDVDs because it has a neat feature, it can boot directory from an ISO image stored on the DVD, so you get to boot the distro from the DVD or burn your own disc from the ISO file. So far so good, but the distros boot system must support this, and many do.

There are various ways of doing this, involving passing arguments with the boot command. Different distros use different methods, but they all end up doing the same thing. System Rescue CD's method works well and Ubuntu's procedure works equally well. So far we have change and choice and all is good.

Why would an Ubuntu-based distro throw out the existing, working method and change it for something else, then? Something that's undocumented and, even after trying all suggested options, simply didn't work? I won't name the distros, but it's not on this month's DVD. Developers, please make things better for us or easier for you, but not just because you can.

Neil

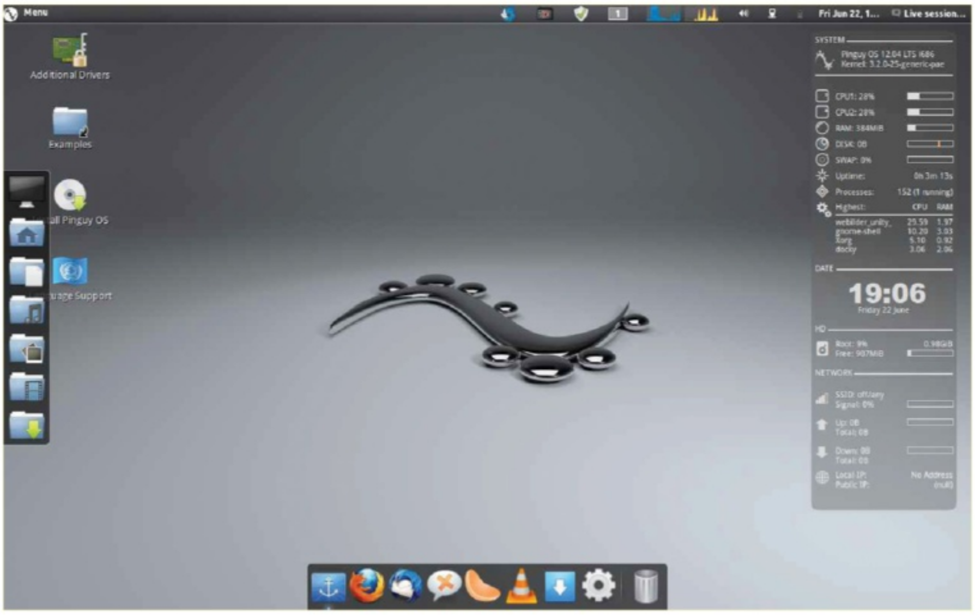
» Important NOTICE!

Defective discs

In the unlikely event of your *Linux Format* coverdisc being in any way defective, please visit our support site at **www.linuxformat.com/dvdsupport** for further assistance. If you would prefer to talk to a member of our reader support team, email us at support@futurenet.co.uk or telephone +44 (0) 1225 822743.

Linux distribution

Pinguy



We have compared a host of distributions that are suitable for new users in this month's Round up, so it seemed logical to include at least a couple of the candidates on the DVD. We couldn't include all of them, as some are huge and would require us switching to a Blu-Ray disc for the covermount.

Pinguy was well received as an Ubuntu derivative that's easy to get to grips with, so here it is. It's a lot larger than a standard Ubuntu CD, meaning if you're after a ready to go operating system, you should have all the software you need to get started without looking for extra packages.

Linux distribution

elementary OS

The other 'new user friendly' distro on the DVD is Elementary OS, which has a desktop that will be familiar to users of Mac OSX. We realise that not all of our users are new to Linux (otherwise we'd have to wonder who was buying the magazine before this year) but a distro that is good for new users can also be good for everyone else.

After all, who doesn't want a distro that installs easily, detects your hardware and lets you get on with what you want to do without extra effort? If your answer to that question is 'me' and you want something hard core that boots to a console by default, we have System Rescue CD. This is one of the best live CDs out there for system repair, it comes with 32-bit and 64 bit kernels and a plethora of tools needed to repair or setup Linux and Windows systems.

LINUX FORMAT DVD

DISTROS | SOFTWARE | GUIDES

Elementary OS Luna

The best-looking distro you'll try this year

Pinguy OS 13.04

Want to get started with Linux? Here's the distribution for complete beginners

64 bit

LXF178 Plus Books, code, podcasts, Hotpicks and more!

Open Index.html to start exploring

New to Linux? Start here

» What is Linux? How do I install it?
» Is there an equivalent of *MS Office*?
» What's this command line all about?
» How do I install software?

Open **Index.html** on the disc to find out

Are you reading this on a tablet?

Download your DVD from **www.linuxformat.com**

From the magazine

Round up

Try the stars of our pick of the new user distros

Tutorials

Software for the custom desktop tutorial and more.

Server software

Useful tips and programs for getting High Availability

LXFHotPicks

Checkbashisms

Performs basic checks on shell scripts for the presence of non-portable syntax.

DigiKam

For those seeking an advanced digital photo management application for KDE.

DustRacing2D

A tile-based, cross-platform 2D racing game written in *Qt* (C++) and OpenGL.

Chrzazscz

An open source maze game… featuring Orcs!

Fgallery

A static photo gallery generator with no frills that has a stylish, minimalist look.

Mars Simulation

A free software Java project to create a simulation of future human settlement of Mars.

Thunderbird

The latest spin of the email application that's easy to set up and loaded with great features.

SeaMonkey

The all-in-one Internet application suite.

Bd

Quickly go back to a parent directory in your shell.

CropGUI

A *GTK* program for lossless cropping of JPEG images.

XRoar

A Dragon 32/64 emulator.

Podcasts

TuxRadar

S5 E17 Like the legend of the phoenix

News Barry Kauler, founder of Puppy Linux has retired (again), Valve announce Linux-based SteamOS for its Steam Machines, Nvidia are helping with open source drivers and CyanogenMod is rolling in it.
Open Ballot A head of Steam, or just a fizzle?
Discoveries of the week Ben pinpoints where Hell is located while Mike likes Winning. Graham, Andrew and Effy seem to have spent the last months doing nothing but play games (*Sir, You Are Being Hunted*, *Pixel Junk Monsters* and *Little Master*).
Reader warning: This podcast has a sad ending. Please have the tissues handy!

And more!

System tools

Essentials

Checkinstall Install tarballs with your package manager.

GNU Core Utils The basic utilities that should exist on every operating system.

Hardinfo A system benchmarking tool.

Plop A simple boot manager to start operating systems.

RaWrite Create boot floppy disks in Windows.

SBM An OS-independent boot manager with an easy-to-use interface.

WvDial Connect to the internet with a dial-up modem.

Reading matter

Bookshelf

The Cathedral and the Bazaar Eric S Raymond's classic text explaining the advantages of open development.

Linux Kernel in a Nutshell An introduction to the kernel written by master-hacker, Greg Kroah-Hartman.

Debian Administrators' Handbook An essential guide for sysadmins.

Linux Dictionary The A to Z of everything to do with Linux.

Dive Into Python A masterclass in this popular language.

Bourne Shell Scripting Guide Get started with shell scripting.

LXF 179 will be on sale Thursday 5 December 2013

Next month

Get started with Linux

Grab your loved ones, lock them in a room with this magazine and a PC and don't let them out until they've stopped worrying and learned to love Linux.

After Ada

Free software should be the purest meritocracy imaginable – so where are the women in FOSS?

Games

Is Linux is going to be the saviour of gaming? We investigate and get very, very excited.

Football Manager

We're not claiming that this is the best football management game in the world, but it's in the top 1.

Contents of future issues subject to change – we might be watching Brian Clough interviews on YouTube.

LINUX FORMAT
#1 for Free Software

Future Publishing Ltd, 30 Monmouth Street,
Bath BA1 2BW **Tel** 01225 442244
Email linuxformat@futurenet.com

abc **21,784** January – December 2012
A member of the Audited Bureau of Circulations.

EDITORIAL
Editor Graham Morrison
graham.morrison@futurenet.com
Deputy editor Andrew Gregory
andrew.gregory@futurenet.com
Operations editor Chris Thornett
chris.thornett@futurenet.com
Art editor Efrain Hernandez-Mendoza
efrain.hernandez-mendoza@futurenet.com
Editorial contributors Jono Bacon, Neil Bothwick, Chris Brown, Tony Chapman, Matthew Hanson, David Hayward, Juliet Kemp, Les Pounder, Jonathan Roberts, Daniel Samuels, Mike Saunders, Mayank Sharma, Shashank Sharma
Art assistance Donna-Marie Scrase
Illustrations Shane Collinge, Simon Middleweek, Elly Walton Illustrations

ADVERTISING SALES
Advertising sales director Nick Weatherall
020 7042 4155 *nick.weatherall@futurenet.com*
Account sales manager Ricardo Sidoli
020 7042 4124 *ricardo.sidoli@futurenet.co.uk*

MANAGERIAL & MARKETING
Campaign manager Sam Book
Head of Computing Group Ian Robson
UK chief operating officer Mark Wood

PRODUCTION AND DISTRIBUTION
Production co-ordinator Ian Wardle
Production manager Mark Constance
Trade marketing manager Colin Hornby
Distributed by Seymour Distribution Ltd, 2 East Poultry Avenue, London EC1A 9PT **Tel** 020 7429 4000
Overseas distribution by Future Publishing Ltd
Tel +44 (0)1225 442244

LICENSING
Licensing & syndication manager Regina Erak
regina.erak@futurenet.co.uk **Tel** +44 (0)1225 732359

CIRCULATION
International account manager Rebecca Hill
rebecca.hill@futurenet.com

SUBSCRIPTIONS & BACK ISSUES
UK readers' order line 0844 848 2852
General UK enquiries 0844 848 1602
Overseas readers +44 (0)1604 251045
Online enquiries www.myfavouritemagazines.co.uk
Email *linuxformatdvd1@subscription.co.uk*

ADDITIONAL MANAGEMENT
Editorial director Jim Douglas
Creative director Bob Abbott
Group art editor Steve Gotobed

Printed in the UK by William Gibbons.

Future

Future produces high-quality multimedia products which reach our audiences online, on mobile and in print. Future attracts over 50 million consumers to its brands every month across five core sectors: Technology, Entertainment, Music, Creative and Sports & Auto. We export and license our publications to 89 countries around the world.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR).
www.futureplc.com

Chief executive Mark Wood
Non-executive chairman Peter Allen
Chief financial officer Graham Harding
Tel +44 (0)207 042 4000 (London)
Tel +44 (0)1225 442 244 (Bath)

We are committed to only using magazine paper which is derived from well-managed, certified forestry and chlorine-free manufacture. Future Publishing and its paper suppliers have been independently certified in accordance with the rules of the FSC (Forest Stewardship Council).

recycle
When you have finished with this magazine please recycle it.

FSC
Mixed Sources
Product group from well-managed forests and other controlled sources
Cert no. TT-COC-1815
www.fsc.org
© 1996 Forest Stewardship Council